ARMY RESEARCH LABORATORY

Modeling Intelligence Production Performance

Marsha B. McLean
Beverly G. Knapp

ARL-CR-444 SEPTEMBER 1999

prepared by

BDM Federal, Inc.
4067 Enterprise Way
Sierra Vista, AZ 85635

under contract

DAAL01-94-C-0048

Approved for public release; distribution is unlimited.

Visual C++® is a registered trademark of Microsoft.

Windows™ is a trademark of Microsoft Corporation.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5425

ARL-CR-444 September 1999

Modeling Intelligence Production Performance

prepared by

Marsha B. McLean
BDM Federal, Inc.

Beverly G. Knapp
Human Research and Engineering Directorate, ARL

under contract

DAAL01-94-C-0048

Abstract

The objective of this effort was to develop an analysis framework and computer-based tool for simulating and evaluating the impacts of materiel, organizational, and personnel changes in the military intelligence (MI) production system. This tool was designed to assist the MI community in assessing new concepts for meeting commander's intelligence requirements of the future.

A series of representational models was built first: conceptual, performance, and information quality. The Conceptual Model represented intelligence production as a simple input-process-output model, with nodes representing the functions required to produce intelligence and links representing the information flow. The Performance Model specified the behavioral tasks required to produce intelligence, taxonomy of human performance errors associated with the tasks, and the operational, scenario, and environmental variables that affect task performance. Finally, the Intelligence Quality Model quantified the results of information flow activity and linked the impact of task performance variables when operating on the information.

A team of experts in behavioral science, modeling and simulation, and military intelligence built the Intelligence Production Model (IPM). The computer-based IPM was then built by linking these models using a rule-based logic structure and was accessed by a user interface designed to allow analysts to conduct case studies for a wide range of evaluation questions. The IPM runs in a Windows™-based PC environment and is being applied to a number of questions raised by the MI operational community.

ACKNOWLEDGMENTS

The authors would like to convey their sincere gratitude and appreciation to the people who contributed their time and effort to make this project possible, especially Dr. Julie Hopson, Dr. David Burnstein, Dr. John Warner, Fran Ainslie, Frank LaNasa, Angela Roe, and Jamieson Christian.

CONTENTS

MODELING INTELLIGENCE PRODUCTION PERFORMANCE

INTRODUCTION

Political changes and technology advances impact all aspects of today's military. The Army is meeting the challenge with reorganization and modernization of its forces through Force XXI. Force XXI is a concept for the evolution of full-dimensional operations for the Army of the early 21st century (Training and Doctrine Command, 1994). In constructing a vision for the future, the Army emphasizes the importance of information and limitless application of information technology.

The proliferation of information and information technologies, coupled with budgetary constraints, offers particular challenges to military intelligence (MI). Information has always been the currency of MI; these challenges serve only to heighten an existing focus. It is critical to understand how projected organizational and materiel changes will impact MI. Ideally, that understanding would precede the implementation of change; computer modeling offers a cost-effective way to assess the impact of change in intelligence production before implementation. One measure of the effect of the change is value added or removed, for example. If a new data processor were being used, an organizational change, value added would be the difference between output using the new processor and output not using it. The performance would be measured in terms of effectiveness and efficiency (Burnstein, 1994).

This report describes the development of a conceptual and computer model of the military intelligence production system called the Intelligence Production Model (IPM). The purpose of the computer model and its underlying concepts is to assist combat, doctrine, and training developers in assessing the impact of change on the processing and quality of information. Using the model as a diagnostic tool, that is, comparing the output of two case studies during changed conditions, the developer can evaluate the impact of change and can formulate solutions regarding training, functional, and organizational aspects of the intelligence production system. The IPM can also be used as a tool to predict as well as evaluate.

OBJECTIVE

The objective of this effort was to develop an analysis framework and computer-based tool for describing, simulating, and evaluating the impacts of materiel, organizational, and personnel changes on the intelligence production system. The framework would be used to

identify areas of information processing and production deficiency and to assess possible remedies. The computer model enables us to systematically investigate the effects of variables on organizational performance and then predict and evaluate how deficiencies in information and the use of information impact intelligence production. It was also desirable for this model's results to be generalizable, that is, free of domain content imposed by situational context or intelligence operations and low resolution.

METHOD

The framework for the IPM was designed and developed by the Fort Huachuca Field Element, Human Research and Engineering Directorate, of the U.S. Army Research Laboratory (ARL) over a number of years using a team consisting of research psychologists, modeling engineers, and MI subject matter experts (SMEs). Development of the model framework before the computerization of the model occurred in several phases: the Conceptual Model, the Performance Model, and the Information Quality Model. Model, in this context, refers to some aspect of the intelligence production process as it was represented in each phase.

The MI Conceptual Model

The original Conceptual Model characterized intelligence production as a simple input-process-output model (see Figure 1). The conceptualization was essentially a network model of the intelligence production process, where *nodes* represent the functions or tasks required to produce intelligence, and the *links* between the nodes represent the information product produced at that node and passed to a subsequent function. The "*delta a*" notation on the link from Node A to Node B, for example, represents transformed information, that is, information or data that have been imparted context and meaning by the task performed at Node A.

From this concept followed the decomposition of the functions to a generic task level. The decomposition identified some of the key dimensions of intelligence tasks: the task information requirements, the behavioral components of the task, the procedural and content knowledge required to perform the task, the environment in which the task is being performed, and individual operator characteristics. For a detailed description of the Conceptual Model, see Appendix A. This Conceptual Model provided the point of departure for identifying the independent variables that affect task and system behavior variables, the dependent variables that provide for measures of effectiveness and performance measures, and the task sequencing necessary to simulate human performance in the intelligence production process.

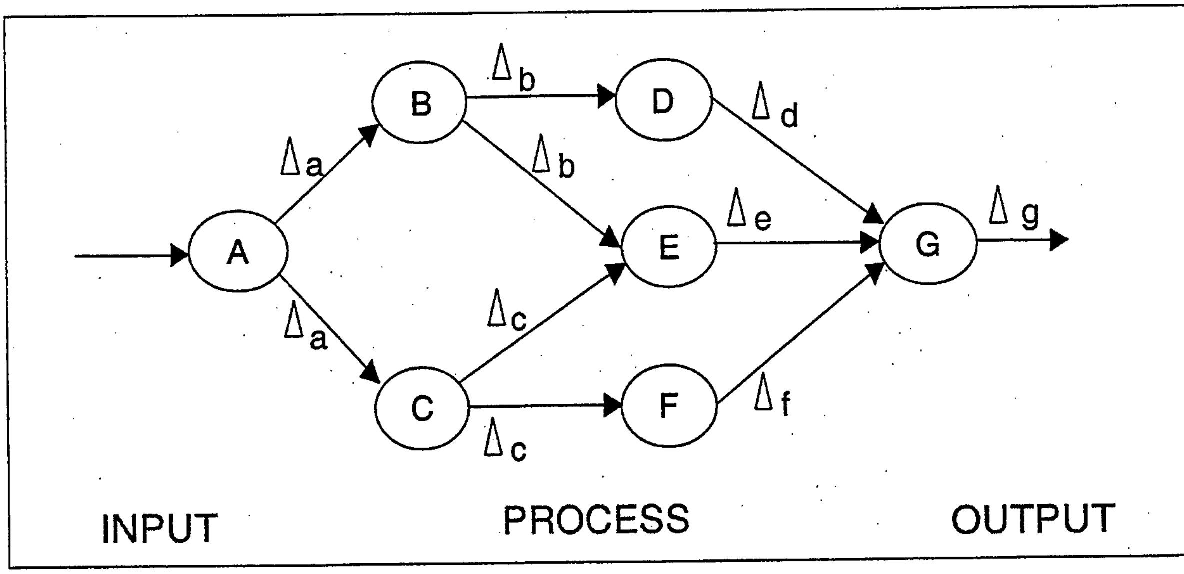

Figure 1. The MI conceptual model.

The MI Performance Model

In order to model intelligence production, it was also necessary to define a *behavioral framework* for capturing intelligence production performance in a computer environment; these behaviors were identified in the decomposition of tasks. The first step was to describe the behaviors to be simulated and to identify the parameters influencing those behaviors. A system performance model was required that emphasized the behavioral aspects of MI production. This model employed an error taxonomy and framework developed for this purpose, which identified the conditions that cause errors. The *error framework* also included a way to assess the impact of error in intelligence production performance.

The next step was to develop a Functional Model that would set the operational context for the Conceptual Model in the MI domain. The Functional Model consists of the functions required to produce intelligence and their decomposition. It was developed independent of any particular MI operational structure by SMEs with reference to current MI doctrine. Psychologists in consultation with SMEs determined behavioral aspects of the decomposition. The error framework and Functional Model are described in detail in Appendix B.

The final step was to develop a Logical Model of intelligence production, which represented the integration of the functional decomposition and error framework within the constraints and assumptions defined for a computer model. The Logical Model described what the model does, the information required to do it, and how the information is used. It was

developed to ensure that all events, information, and rules necessary to computerize the Performance Model had been identified. The Logical Model is described in detail in Appendix C.

The MI Information Quality Model

The final component required to model the MI system was a methodology for representing the intelligence output produced by that system. A methodology for assessing the effectiveness of MI units (Burnstein, Fichtl, Landee-Thompson, & Thompson, 1990) provided the basis for modeling information quality, that is, the measure of how well the intelligence product met the needs of the user of intelligence. This methodology focused on the information requirements of intelligence users and provided a means to diagnose deficiencies in intelligence products. Here, quality was defined as utility to the (intelligence) user as represented by the difference between the required value of information and the value of the information received.

Just as the MI Performance Model requires a functional representation of the MI domain, the MI Information Quality Model requires an informational representation of the MI domain. This representation was developed using a technique called conceptual mapping (Warner & Burnstein, 1996) to build a hierarchical representation of domain knowledge. The Quality Model is portrayed in a structure called the Intelligence Conceptual Map (ICM). The ICM is a normative representation of the MI domain comprised of information entities (nodes) connected to one another in a coherent hierarchy associating all nodes that provide understanding to one another in either parent-child relationships or transformation relationships. As one moves from the lower levels of the map through the higher levels, one goes from detailed, specific data and information to more conceptually oriented general understanding or *knowledge*.

The sample MI conceptual map in Figure 2 illustrates this bottom-up hierarchy, and the relationships between information domains, that is, information represented in Nodes C, L, and M, support and explain each other; information represented in Nodes M and J is transformed into information in (parent) Node F. The actual ICM (see Appendix C) was painstakingly constructed by MI SMEs and represents the development of understanding about future enemy activities, the ultimate MI goal. Understanding of these enemy activities was decomposed into three main hierarchies: friendly and enemy activities and the physical environment, that were further decomposed into progressively lower levels of detail, shown as data elements in Figure 2.

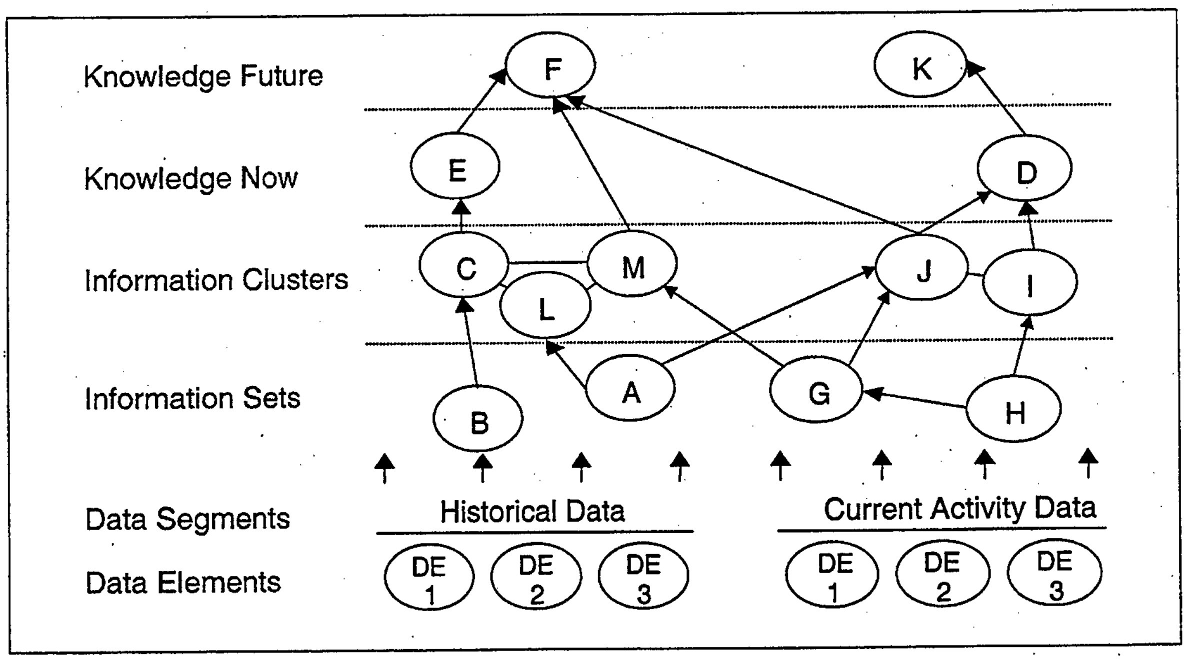

Figure 2. Sample MI conceptual map.

THE IPM COMPUTER MODEL

The purpose of these conceptual models was to guide development of a computer simulation model of intelligence production, to be used as a tool to predict and evaluate changes in the MI production system. The scenario and operational context in which the system is situated drive change in the MI production system. In order to computerize the performance and information quality models, the scenario is used as the mechanism by which the IPM user defines the operating conditions and changes the parameters of interest. Scenario definition included mission, environmental, organizational, collection, personnel, and task parameters that establish the conditions simulated by the IPM. The computer-based IPM was built using a process where the model components were successively developed then linked. During the early phases of software implementation, the representational model was simply computerized. As each model concept matured, other Functional Model components were defined and developed, which served as the link or integrating component of the original representational models.

Functional Performance Model

The Performance and Information Quality Models were originally designed and computerized as stand-alone modules. The Performance Model was implemented first. In addition to the performance variables used to define operator behavior and environment and task

conditions and environment, the Performance Model used a single variable that was a placeholder for collection activities and information quality. This variable was called the *information variable* and represented the systemic value of information, or the value of information derived from the battlefield situation and the operational environment being modeled. Another set of operational variables provided scenario context but only to the extent that they impacted the personal and task environment variables of the intelligence analysts being modeled. Figure 3 depicts the Functional Performance Model.

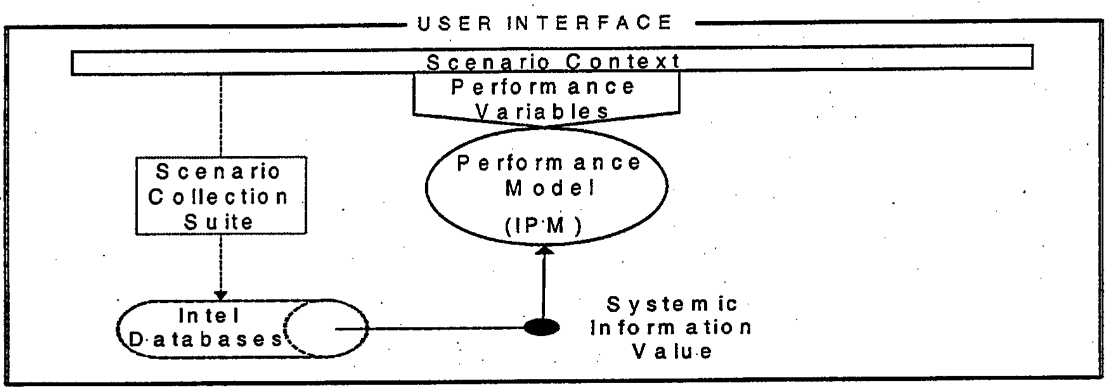

Figure 3. Functional performance model.

Functional Information Quality Model

The Information Quality Model (IQM) was computerized next. The IQM model's information was characterized in the MI production system both in terms of information as it is represented in the Performance Model and as it is represented in the MI conceptual map. With regard to the Performance Model, the IQM expanded and operationalized information value, as shown in Figure 3; this value was treated as a single variable that represented information in the system only marginally. Therefore, one requirement was to simulate collection activities by modeling the collection assets indicated by the operational context of the scenario. In keeping with the context-free approach to information representation, collection activities were instantiated in terms of the operational and battlefield environment and were represented in terms of their contribution to information value in the conceptual map. In Figure 4, the scenario context describes the mission and operational situation, which in turn defines the scenario collection suite that is populating the intelligence databases that instantiate information value in the ICM. The ICM, shown in Figure 4, contains all the information described in terms of value dimensions attributable to collection activities. This information is used by the analysts and their tasks being

modeled in the Performance Model, and the value of this information is conveyed to and interpreted by the IPM in the same manner as the information variable discussed earlier.

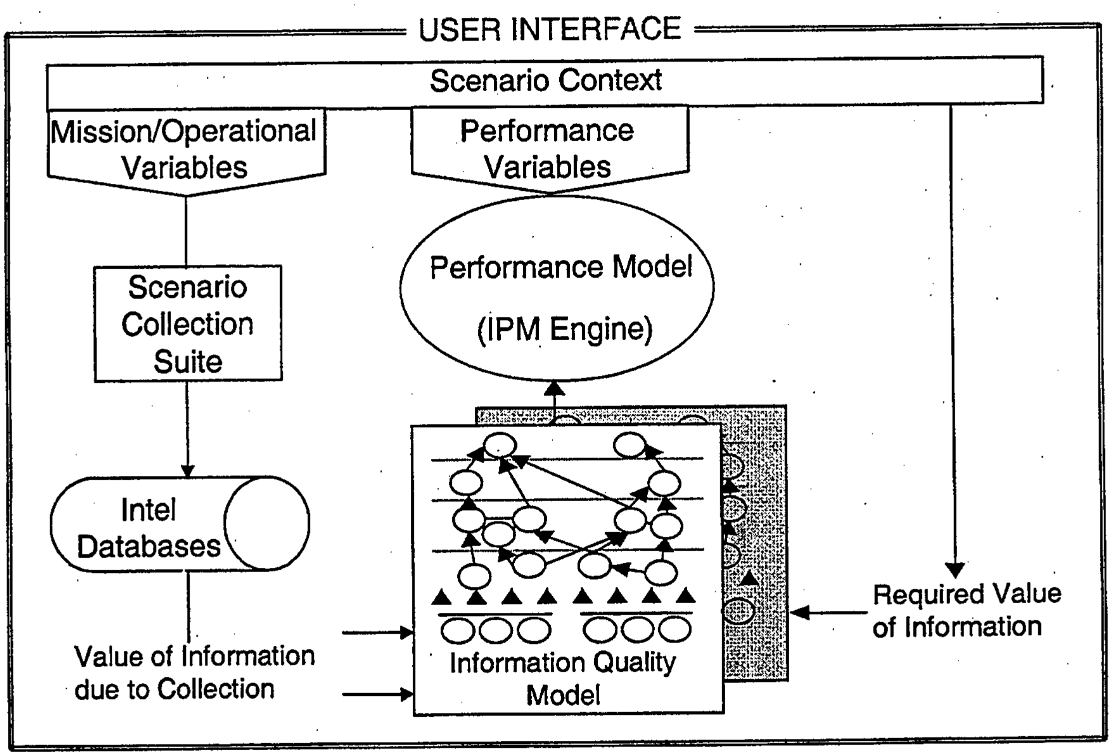

Figure 4. Functional information quality model.

The second requirement was to represent the impact of the analysts' activities on this information. As discussed earlier, the impact of errors committed in the performance of these tasks is instantiated in the Performance Model in terms of their impact on the quality of information being produced.

The final requirement was related more to assessment of the results, or output, of a modeling exercise and is discussed here in terms of its relationship to the scenario context. The purpose of this design was to establish how the scenario context would establish information value requirements. Information quality is a measure of how well the actual value of information met the required value of information. In the MI domain, the required value of information can be identified by the commander's information requirements. In a scenario-driven context, the value of the commander's information requirements is defined with reference to combined parameters such as mission, level of war, battlefield operations, and echelon. Therefore, the value of

information required by the commander (per the scenario) for each node is contained within the ICM nodes, in addition to information value attributable to collection.

Model Integration

The Functional Model

The basic consideration when performance and information quality are integrated was that information quality, the raw material used by intelligence analysts to produce intelligence, may affect the performance of human analysts, and the performance of human analysts may affect the quality of information. The first integration design consideration was how to model this recursive relationship. It was recognized that the Performance Model alone assumed ideal information and the IQM alone assumed ideal performance. The modeling approach chosen was to begin with the value of information collected by the scenario assets (assuming ideal performance), pass this first indication of information value to the Performance Model (in place of the single systemic information variable), execute the Performance Model, and then adjust the value of information, based on human performance. This defines a third instantiation of the ICM, one that contains the value of information in each node attributable to the combination of collection and analysis, that is, the value of information based on human performance. Figure 5 shows all elements of the integrated Functional Models.

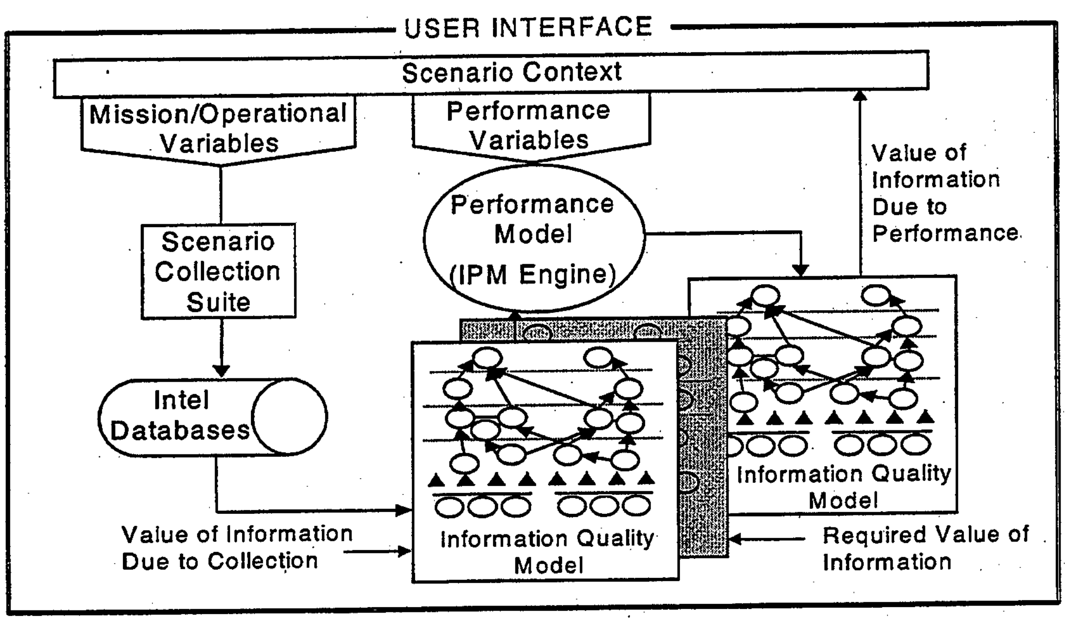

Figure 5. IPM integrated model.

User Interface: Model Input (setup)

Another design requirement for the integrated functional components of the model was to define a "shell" that could be linked to all the internal processing components of the model. This shell serves as the user's interface to the IPM, so that all the scenario context, collection suite, and performance variables can be set by the user before the model is run. Figure 6 depicts this shell and the IPM components from the user's perspective.

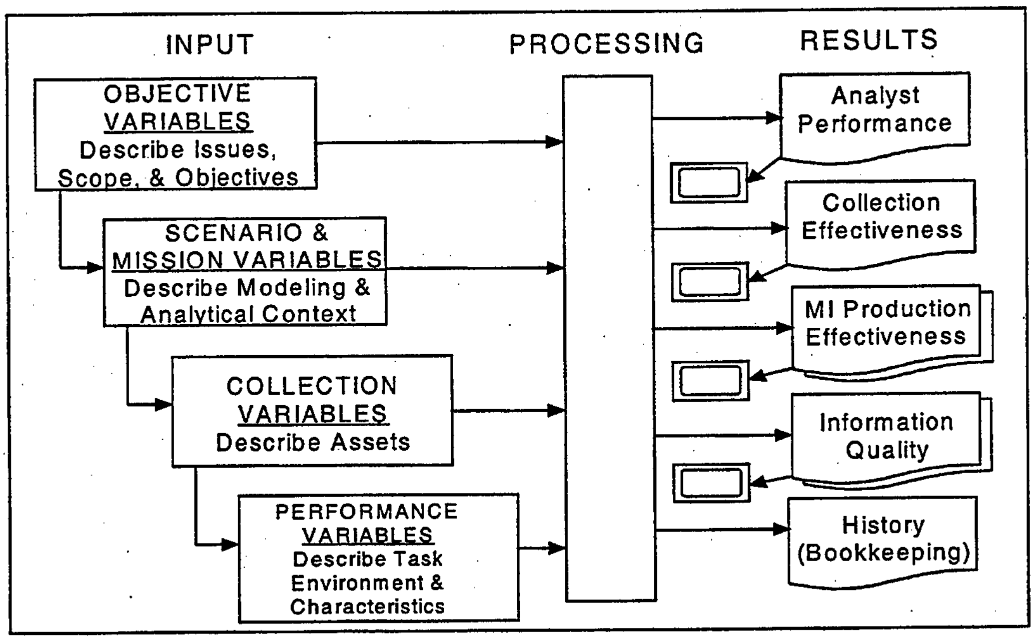

Figure 6. IPM shell and model components.

The interface was designed to be oriented to the user's objectives for employing the model. A query-based framework was developed to aid the user in focusing the model setup on the scope and objective of the modeling session, as well as in formatting output according to the level of detail implied by his or her objective. A user's session begins with an interview or query screen that guides the user in an objective description of the modeling session.

User Interface: Model Output (results)

Each functional component has its own set of textual and tabular output reports that are extensive and extremely detailed listings of information quality and performance "measures." At a more aggregate level, an additional set of reporting utilities is available, which provides graphical displays that can be viewed on the computer screen so the user can obtain a summary of emergent results on line. The graphical output uses a color-coding scheme that

represents information quality (the value of information available versus the value of information required) attributable to collection and analysis, as well as a scheme that provides an overall depiction of analyst performance.

The raw data used to build the graphical and textual reports are also available and may be manipulated in various forms to create specific output representations for analysis. The results of several test cases, against each other or in comparison to a baseline, could be graphed for a single variable, such as value of information collected, numbers of errors, or types of errors. Another example of output data representation could be like a state of performance graph for a single test case in which the final state of several variables could be combined to give an overall measure of performance for the given case being studied.

Running the IPM

The IPM was developed in Microsoft Visual C++® for a Windows™ 95 operating system. It runs in real time on any personal computer (PC) with the following minimum configuration: 486 with 8 megabytes (MB) of random access memory (RAM), 10 MB hard drive space, plus additional space to store test cases as they are run. The IPM mostly follows basic Microsoft graphic user interface protocols and has most of the standard utilities. The IPM does have some very strict file-naming conventions that it enforces when establishing and running test cases.

SUMMARY

A model of the MI production system has been developed, beginning with the creation of descriptive models (the Conceptual Model, the Error Framework, the Functional Model, the Logical Model, and the Information Quality Model) and culminating in an executable computer model. This analytical tool runs on a stand-alone PC and features an extensive rule base and set of variables and parameters that allow comprehensive evaluation of the impact of system and scenario factors on intelligence production performance.

To date, the IPM has been used both in validation and analytical exercises. The validation exercises were performed for the purpose of (a) determining the "face validity" of the model logic and output and (b) addressing some specific questions related to the domain. These exercises included a recreation of the Grenada operation, investigations of collection asset trade-offs for the Directorate of Combat Developments, U.S. Army Intelligence Center, Fort Huachuca, Arizona, and investigations of information warfare issues (information degradation, deception, etc.). For

the analytical exercise, the IPM was modified to reflect the scenario and operational environment of the DIV XXI Advanced Warfighting Experiment conducted at Fort Hood, Texas. Under the direction of the Battle Command Battle Lab, excursions were then planned and conducted, which would assess performance and identify issues for the "digitized" environment of the 21st century Army.

Future activities should include enhancements of the IPM that fully model and instantiate intelligence production in the digital environment, including collection pre-processors, evolving approaches to collection and target management, and the impact of computers, software, and increased information volume on information performance. In addition, investigations are in the process of looking at integrating the IPM with other intelligence and human performance models. One such concept is to integrate the IPM with the Task-Network Modeling to produce a "C3I Meta-Model." The C3I Meta-Model would enable a user to assess intelligence production performance from both a work flow and a logical (or rule-based) perspective. In other words, this combined model could be used to investigate both efficiency and effectiveness in intelligence production systems. Outside the MI domain, much of the research and findings leading to the development of the representational models could be adapted to information processing in other domains, both military and non-military.

REFERENCES

Al-Kunifed, A. & Wandersee, J.H. (1990). One hundred references related to concept mapping. Journal of Research in Science Teaching, 27(10), 1069-1075.

Altman, J.W. (1966). Classification of human error, in Symposium on the Reliability of Human Performance in Work, W. B. Askren (Ed.). Annual Convention of American Psychology Association.

Ausubel, D.P. (1968). Educational psychology: A cognitive view. New York: Holt, Rinehart and Winston.

Baars, B.J. (1980). On eliciting predictable speech errors in the laboratory. In V. A. Fromkin (Ed.), Errors in linguistic performance: Slips of the tongue, ear, pen, and hand (pp. 308- 318). New York: Academic Press.

Bagnara, S., Rizzo, A., & Stablum F. (1989). Error analysis in man-machine systems. Paper presented at the meeting of the Sixth Euredata Conference: Reliability, data collection, and use in risk and availability assessment. Siena, Italy.

Bernard, R.M. & Naidu, S. (1992). Post-questioning, concept mapping, and feedback: A distance education field experiment. British Journal of Educational Technology, 23(1), 48-60.

Brehmer, B. (1987). Models of diagnostic judgements. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), New technology and human error (pp. 87-95). New York: John Wiley & Sons.

Burnstein, D.D. (1994). Personal communication, based on a work in progress.

Burnstein, D.D., Fichtl, T., Landee-Thompson, B., & Thompson, J. (1990). Implementation guide for assessing intelligence production effectiveness (ARI Research Product 30-92). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Cossette, P., & Audet, M. (May 1992). Mapping of idiosyncratic schema, Journal of Management Studies, 29:3, 325-347.

Donald, J.G. (1987). Learning schemata: Methods of representing cognitive, content, and curriculum structures in higher education. Instructional Science, 16, 187-211.

Eden, C. (1994). Cognitive mapping and problem structuring for system dynamics model building. System Dynamic Review, 10(2-3), 257-276.

Finley, F.N. & Stewart, J. (1982). Representing substantive structures. Science Education, 66(4), 593-611.

Gawron, V.J., Drury, C.C., Czaja, S.J., & Wilkins, D.M. (1989). A taxonomy of independent variables affecting human performance. International Journal of Man-Machine Studies, 31, 643-672.

Gillan, D.J., Breedin, S.D., & Cooke, N.J. (1992). Network and multidimensional representations of the declarative knowledge of human-computer interface design experts. International Journal of Man-Machine Studies, 36, 587-615.

Gordon, S.E., Schimierer, K.A., & Gill, R.T. (1993). Conceptual graph analysis: Knowledge acquisition for instructional system design. Human Factors, 35, 459-481.

Hale, A.R. (1990). Safety rules O.K.? Possibilities and limitations in behavioral safety. Journal of Occupational Accidents, 12, 3-20.

Hartley, R.T., Coombs, M.J., & Dietrich, E. (1988). An algorithm for open-world reasoning using model generation (CRL Memoranda Series MCCS-88-123). Las Cruces, NM: New Mexico State University.

Lambiotte, J.G., Dansereau, D.F., Cross, D.R., & Reynolds, S.B. (1989). Multirelational semantic maps. Educational Psychology Review, 1(4), 331-367.

Leong, T. (1992). Representing context-sensitive knowledge in a network formalism: A preliminary report. Proceedings of the 8th Conference of Uncertainty in AI, Stanford.

Michon, J.A., Smiley, A., & Aasman, J. (1990). Errors and driver support systems. Ergonomics, 33(10/11), 1215-1229.

Miller, D.P. & Swain, A.D. (1987). Human error and human reliability. In G. Salvendy (Ed.), Handbook of human factors (pp. 219-250). New York: John Wiley & Sons.

Muckler, F.A., Seven, S., & Akman, A. (1990). Proposed method for military intelligence job ability assessment (ARI Research Note 90-135). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Navarro, C. (1989). A method of studying errors in flight crew communication. Perceptual and Motor Skills, 69, 719-722.

Norman, D.A. (1981). Categorization of action slips. Psychological Review, 88(1), 1-15.

Norman, D.A. (1983). Design rules based on the analysis of human error. Communications of the ACM, 26(4), 254-258.

Novak, J.D. & Gowin, D.B. (1984). Learning how to learn. New York: Cambridge University Press.

Rasmussen, J. (1980). What can be learned from human error reports. In K. D. Duncan, M. M. Gruneberg, & D. Wallis (Eds.), Changes in working life (pp. 97-113). New York: John Wiley & Sons.

Rasmussen, J. (1982). Human errors. A taxonomy for describing human malfunctions in industrial installations. Journal of Occupational Accidents, 4(2-4), 311-335.

Rasmussen, J. (1986). Information processing and human machine interaction: An approach to cognitive engineering. New York: North Holland.

Rasmussen, J. (1987a). The definition of human error and a taxonomy for technical system design. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), New technology and human error (pp. 23-30). New York: John Wiley & Sons.

Rasmussen, J. (1987b). Cognitive control and human error mechanisms. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), New technology and human error (pp. 53-61). New York: John Wiley & Sons.

Reason, J. (1986). Recurrent errors in process environments: Some implications for the design of decision support systems. In E. Hollnagel, et al. (Eds.), Intelligent decision support in process environments (pp. 255-270). NATO ASI Series, Vol F21. Berlin: Springer- Verlag.

Reason, J. (1987a). A framework for classifying errors. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), New technology and human error (pp. 5-14). New York: John Wiley & Sons.

Reason, J. (1987b). A preliminary classification of mistakes. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), New technology and human error (pp. 15-22). New York: John Wiley & Sons.

Reason, J. (1987c). The psychology of mistakes: A brief review of planning failures. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), New technology and human error (pp. 45-52). New York: John Wiley & Sons.

Reason, J. (1987d). Generic error-modelling system (GEMS): A cognitive framework for locating common human error forms. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), New technology and human error (pp. 63-83). New York: John Wiley & Sons.

Reason, J. (1987e). Cognitive aids in process environments: Prostheses or tools? International Journal of Man-Machine Studies, 27, 463-470.

Reason, J. (1990). Human error. Cambridge: Cambridge University Press.

Rigby, L.V. (1970). The nature of human error. In Annual technical conference transactions of the ASQC (pp. 457-466). Milwaukee, WI: American Society for Quality Control.

Rouse, W.B. (1990). Designing of human error: Concepts for error tolerant systems. In H. R. Booher (Ed.), MANPRINT: An approach to systems integration (pp. 237-255). New York: Van Nostrand Reinhold.

Senders, J.W. & Moray, N.P. (1991). Human error: Cause, prediction, and reduction. Hillsdale, NJ: Lawrence Erlbaum Associates.

Training and Doctrine Command (1994). Force XXI operations (TRADOC Pamphlet 525-5). Ft. Monroe, VA: Author.

Warner, J. & Burnstein, B. (1996). Situation, domain and coherence: Towards a pragmatic psychology of understanding (ARL Technical Report ARL-CR-306). Aberdeen Proving Ground, MD: U.S. Army Research Laboratory.

Woods, D.D. (1989). Modeling and predicting human error. In J. I. Elkind, S. K. Card, J. Hochberg, & B. M. Huey (Eds.), Human performance models for computer-aided engineering. Washington, DC: National Academy Press.

APPENDIX A

INITIAL CONCEPTUALIZATION

INITIAL CONCEPTUALIZATION

This appendix presents a conceptual model of intelligence production. The model is the basis for a computer simulation model of intelligence production that includes the human components.

NEED FOR A MODEL

Recent advances in technology have resulted in an explosion of information available for intelligence production. In addition, projected technological innovations for collecting data will add to the information load. The result is an increased processing load on intelligence analysis, the most human-intensive function.

Significant efforts are under way to solve the challenges of processing the mountains of information into intelligence. Processors can integrate the performance parameters of collectors and provide an initial fusion of information. However, little has been done to determine how the human component of the intelligence production system is affected by new technology. As new systems are fielded, an understanding of the human component of intelligence production is essential for maximizing the performance of the intelligence system.

In austere budgetary conditions, it is crucial to understand the impact of changing doctrine, decreased resources, expensive new systems, and realignment of the intelligence structure. This is because changes must not result in degraded intelligence. Therefore, changes in the intelligence production system need to be assessed before implementation, preferably in the earliest stages. Computer modeling and simulation is the most cost-effective way to make the initial assessments and evaluate proposed implementations.

IMPETUS FOR MODELING

A methodology for assessing the effectiveness of MI units had previously been developed. It identified the information requirements of the intelligence users and a procedure to determine their information priorities. In addition, users identified the dimensions of intelligence and set standards for its acceptability. The methodology provided a way to identify both the strengths and weaknesses of intelligence units. Included, based on the user's assessment of the intelligence received, was a method for diagnosing production deficiencies. The strategy used was a fault analysis built upon backward engineering and the development of the intelligence product.

It was with the understanding of how to assess intelligence production in the field that a concept of how to produce a computer simulation model began. A computer model that could build upon the backward engineering of the diagnostic strategy would provide a structure to assess changes in the production system. In addition, a model that can simulate change could be used to predict as well as evaluate.

MILITARY INTELLIGENCE IN PERSPECTIVE

MI represents a system of systems. Its goal is to describe and provide insights about an enemy or potential enemy. No matter how much or how little information is available, sufficient and cogent intelligence must be produced.

While many individuals and organizations use information to produce products, MI is an example of a pure information production organization. MI requires raw data and processed information as its raw materials. Its functions transform the raw material into intelligence, both descriptive and predictive. As an organization, MI is large and complex. It requires many people working in different locations, at many organizational levels, using a variety of equipment of different complexity to produce various tailored outputs. A generalized list of the characteristics of the MI system is shown in Table A-1.

HUMAN PERFORMANCE IN PERSPECTIVE

Most behavioral research about using information has focused on how individuals process information rather than how organizations use and produce information. This focus is evident in research about MI. Although MI is a large and complex information management and processing system, research has primarily focused on soldier functions.

Implicit in the individual approach is that changes to enhance soldier performance will benefit the system. While this may be true for a specific task, it may not be true for the entire system. It may also be true that change meant to enhance the performance of the MI system may damage soldier performance and that system performance is actually degraded. In MI, the interaction of individuals and equipment contributes to the system output. This means we must know how a change in one part of the system affects the entire system. Thus, the effective prediction, diagnoses, and modification of human performance depend upon relationships within the system. Any changes that may improve individual performance must be compatible with the system's requirements and vice versa.

Table A-1

Characteristics of the MI System

User Relationships	Supports a hierarchical Army command structure by providing intelligence (processed information) both descriptive and predictive. Provides a structured hierarchy that supports the equivalent level of the command structure. The output of each level of the MI hierarchy is tailored to the needs of the command structure at that level.
Input-Output Considerations	Information (processed and unprocessed) is used as its raw material and produces processed information as its final output. Different levels of the MI hierarchy can receive common information or information unique to that level of the hierarchy. Within the hierarchy of the MI structure, either unprocessed information (raw data) or processed information is passed to the next higher level. Within the hierarchy of the MI structure, only intelligence or combat information is passed to the next lower level in the hierarchy.
Processing Considerations	At any level within the MI structure, intelligence production (the processing of either raw data or processed information) can be described by the production functions required to produce the intelligence. Without respect to the level of the MI hierarchy or functions, each function may require the same information production tasks to be performed, although not to the same degree. These common information production tasks are planning, collecting, managing, analyzing, integrating, interpreting, preparing, and disseminating information. Humans, machines, or a combination depending on the structural level of the MI system and the production functions being performed accomplishes the information production.

THE MI PRODUCTION CONCEPTUAL MODEL

In order to develop a conceptual model, it was necessary to account for the MI characteristics described in Table A-1. At the most abstract level, MI production was considered as a simple input-process-output (I-P-O) model (see Figure A-1). The processes as identified in Figure A-1 represented the basic information transformations. However, to be useful, the model had to be expanded. It needed to depict the complexities of the MI systems while retaining the I-P-O character.

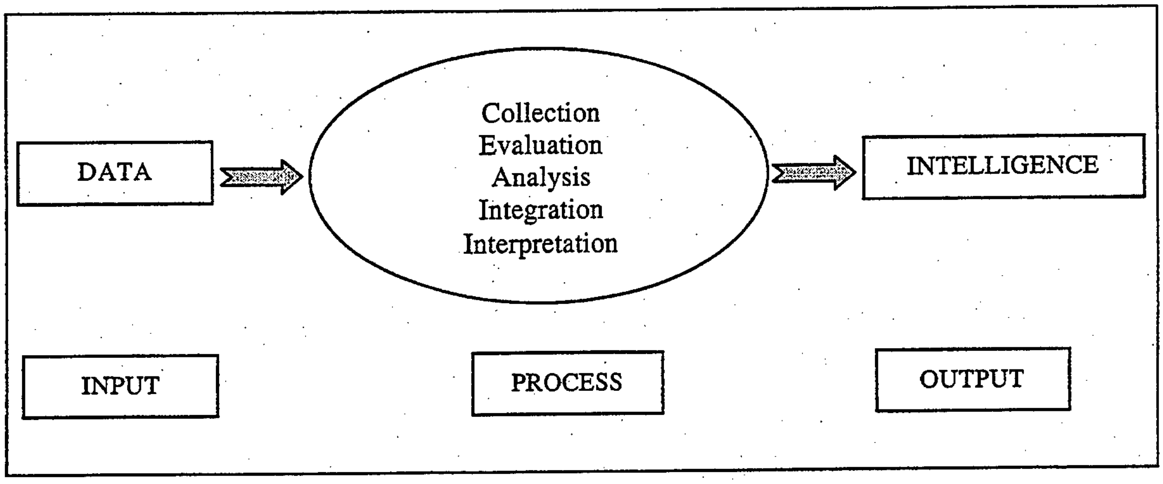

Figure A-1. Intelligence production: A simple I-P-O model.

The best representation of the expansion was a network model. In Figure A-2, the nodes (circles) represent functions required to produce intelligence. Though not shown, the production tasks are nested within the nodes, where the transformation or information occurs. The links (lines with arrows) between the nodes show where the output of the node is used. The product of the nodes is represented by the lettered delta. Deltas were used to emphasize the changing nature of the information as it passes through the production system. The lower case letters on the links describe the path on which information flows in the MI production system. This conceptual model represents the minimum requirements that describe intelligence production. In addition, it covers the structural and functional requirements implied by the intelligence production characteristics described in Table A-1. The structural and functional requirements are shown in Table A-2.

The conceptualization (see Figure A-2) represents one level of the MI system. The conceptual model can be expanded (see Figure A-3) to represent different levels of the command structure supported by MI. The arrows between various echelons indicate that information flows between various levels. The flow does not represent the actual interrelationships between the functions within the different levels.

A more important aspect of the model is the decomposition of the functional nodes. It is within the decomposition that the human components of intelligence production become established. Figure A-4 represents the decomposition of a function to a generic task level. The decomposition identifies some of the key dimensions for describing the task.

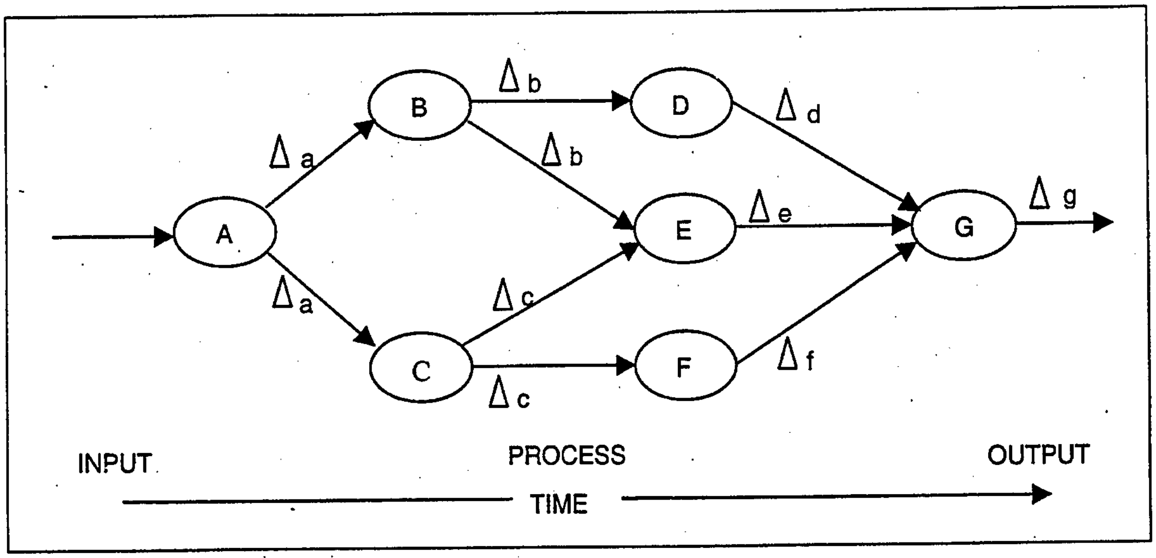

Figure A-2. The MI production conceptual model.

These include

1. The information requirements are the data requirements for the task. Included in the data requirements is their source. The source helps to identify the relationships between functions in the intelligence production hierarchy.

2. The sub-tasks that comprise the task help to identify and clarify other elements of the task description. In addition, they set the boundary for the most detailed level of the decomposition.

3. The dependent variables provide for measures of effectiveness and performance.

4. Procedural and content knowledge specifies the knowledge required to perform the task.

5. The independent variables affect behavior and influence performance. They may be derived from the task (e.g., level of difficulty), the environment in which the task is performed (e.g., workload), or from the operator (e.g., skill level.)

Table A-2

Structural and Functional Requirements of the Conceptual Model

Structurally	Retains the input-process-output characteristics. More than one function (node) is necessary to produce intelligence. Information (intelligence) can flow from one function to another, as depicted by line (link) *ac* or from one function to several functions, as depicted by links *be* and *bd.* Information (intelligence) can be received by one function, as indicated by link *ab*, or received from multiple functions, as indicated by links *ef* and *cf.* The transformation of information to intelligence can follow a simple path, for example links *ac*, *cf*, or a more complex path, for example links *ab*, *bd*, *de*, *ef.* The transformation of information to intelligence occurs in one direction, as indicated by the arrows on the links, and over time, as indicated by the directional arrow labeled time.
Functionally	The nodes contain the intelligence production tasks required to change the input to output (from one delta to another). Each node has a specific output, identified by a lettered delta. The nodal deltas are inputs to other functions. Thus Node B acts to transform output from Node A (delta a) to its own output, delta b, and is then sent to Nodes D and E. Delta g is the final output from the system. Since intelligence production occurs within the node, the output delta must represent the results of production and production performance.

The level of task description depends on the problem being addressed. The decomposition of the conceptual model, as proposed, takes us to the level needed to address organizational effectiveness.

A network model, including both the expansion and the decomposition of the network represents the conceptual model of MI. Although the conceptual model is descriptive, it provides a framework for better understanding the complexities of intelligence production.

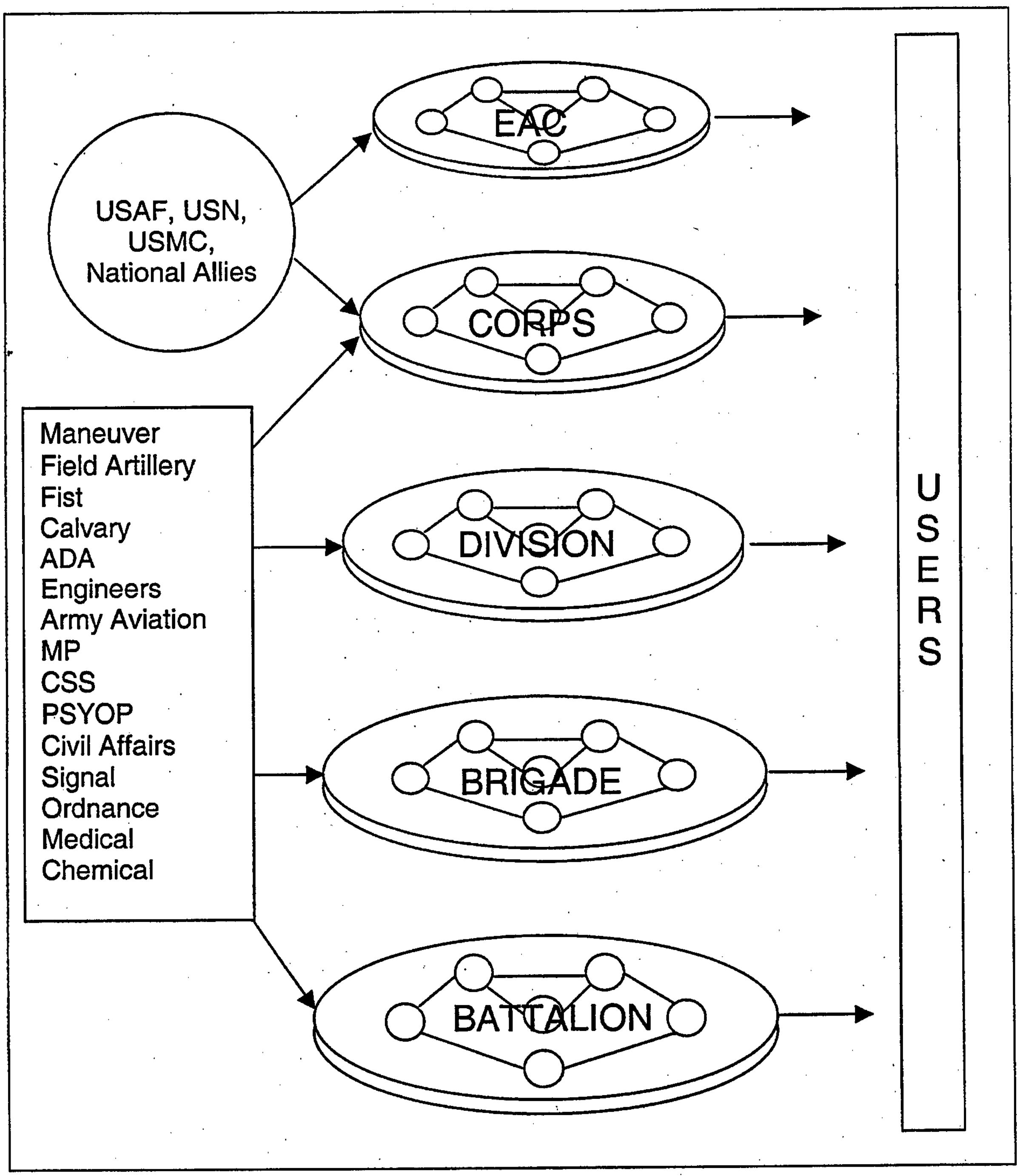

Figure A-3. An expansion of the conceptual model through echelons.

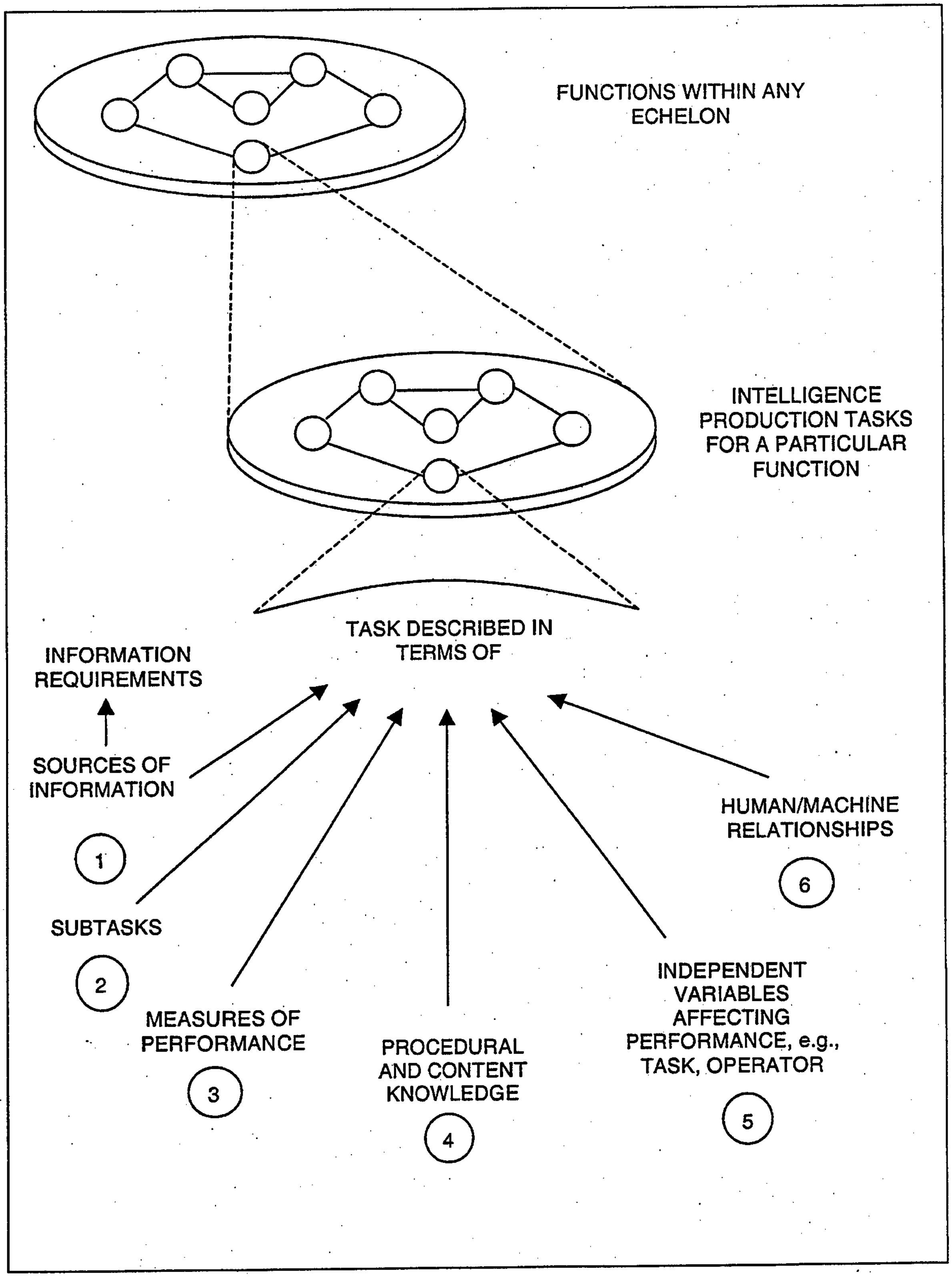

Figure A-4. Decomposition of the conceptual model to a production task level.

MEASURING INTELLIGENCE PRODUCTION

Critical to the modeling effort is the need to measure intelligence production. The ability to measure permits a wide variety of questions to be answered. The conceptual model provided a framework for identifying what, where, and how to measure the elements in the production system. There are three possible measurements: measures of effectiveness (MOE), performance (MOP), and efficiency (MOI).

MOEs are measured against the standards required to perform the next function. The next function could be a node within the MI system or could represent an external user. For example, in Figure A-2, the intelligence user determines the effectiveness of output delta F. The effectiveness of delta A is defined by the requirements of Nodes B and C in the production system. In addition, nodal MOEs must be compatible with the system MOEs. This compatibility is accomplished by an appropriate decomposition of the function in the model. The decomposition enhances the appropriate backward chaining for MOE development. MOEs are the quantity and quality of information.

MOPs are task dependent. They measure the behavior within a node required to produce the output. There are many different MOPs, depending on the type of behavior. Most MOPs within MI can be measured by *time to perform* or *number of errors*.

MOIs are measures of efficiency. They are made within the nodes and represent the cost of changes in performance or effectiveness. Measures include

1. Time costs—it takes a longer or shorter period of time to perform the functions.

2. Manpower costs—the time to perform the function remains the same, but it now takes more or fewer people to perform the function.

3. Error modification—it forces or eliminates errors from occurring within functions.

4. Transmit errors—passes errors onto the next function that were not characteristics before the change.

5. Opportunity costs—the gain or loss of time and resources at one function may have positive or negative consequences for unrelated functions.

6. Psychological costs—changes could result in, for example, increased stress or frustration. These costs are measured independently of their impact on effectiveness or resource costs.

Any deliberate change in the production system is expected to enhance the effectiveness of efficiency. Change could stem from trying to remedy a system dysfunction or an evolutionary enhancement of the system. The impact of change is measured as value added. Value added is a relative concept that can have either a positive or negative value. It requires the comparison of the measurements resulting from the change to be compared to the measurements before the change.

Value added can occur within a node, for example, by making changes in the production tasks or automating the tasks. The value added could be measured either by efficiency in performing the task or by improvement in the output (effectiveness). Value also can be added by changing the input to any node—input of preprocessed information rather than raw data, for example. Again, either efficiency or effectiveness can be measured to value added. Finally, value could be added by changing the path of information as it flows through the production process. In addition, value added can be determined within nodes or at outputs not directly affected by change. For example, a change in the input delta A (see Figure A-2) could be measured as the value added to delta F. Figure A-5 summarizes the measures of value added.

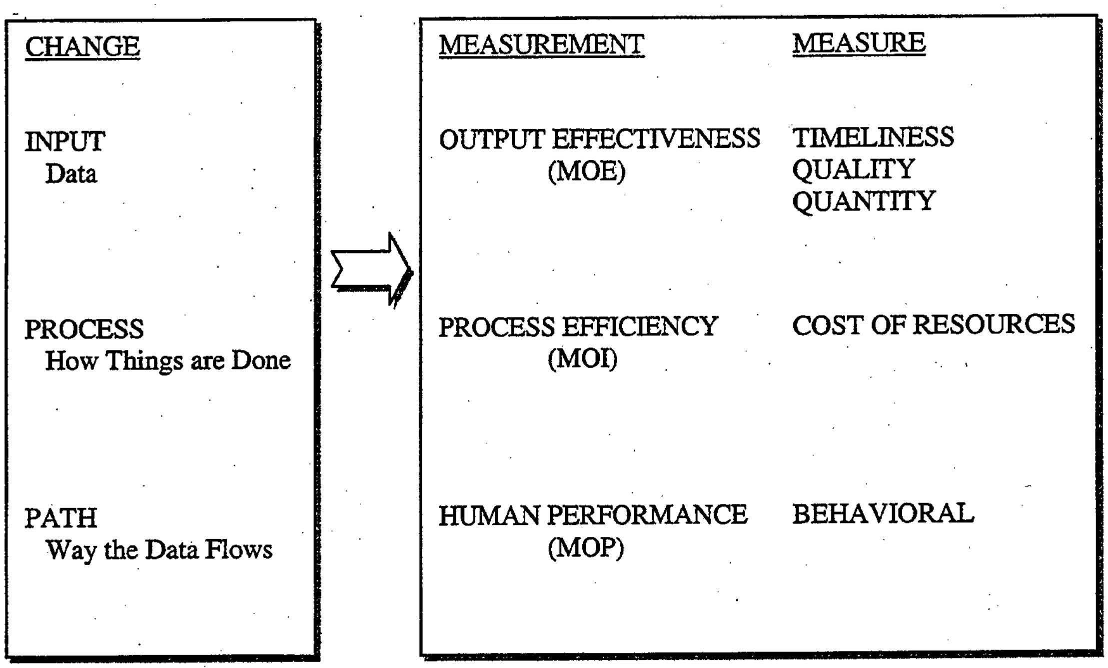

Figure A-5. Measures of value added.

THE MODEL AS A TOOL

The purpose of the conceptual model was to guide the development of a computer simulation model of intelligence production performance. Thus, it must be clear that the

conceptual model could be used as a framework to predict and evaluate changes in the intelligence production system.

Changes include actions taken to remedy a system dysfunction or the implementation of planned modifications (designs) for enhancing efficiency or effectiveness. The direction for change comes from many sources. They include lessons learned during exercises, modifications or doctrinal changes, new training techniques, decreased resources, unit task organization, and imposition of personal preferences. The conceptual model should help to predict and evaluate the impact of the change.

According to the conceptual model, change can be implemented at a node, at input, or along a path. A new standing operating procedure (SOP), the addition of a materiel system, or decrease in resources would be examples of changes implemented within a node. Increasing the amount or kind of information that must be processed by a node would be examples of changes in the input. With reference to Figure A-2, an example of change in the path would be sending information, delta b, directly to Node G rather than Node D.

The effect of any of these changes can be measured at the output of the affected node(s). In the example given, the effect of changing the path is measured as value added (or removed) at Node G. If a new data processor were used (a change within a node), value added would be the difference between output (delta) using the new processor and output not using it, or using the old processor. The performance is measured as effectiveness and efficiency.

The model implies that successful intelligence performance is the result of the adequate functioning of the entire intelligence production system. Therefore, the effect of any change can be measured at all the subsequent deltas in the path. Thus, a change in Node B could result in increased value added to delta B, but the change in delta B as input to Nodes D and E could result in a decrease in the value added of outputs delta D and E. Thus, the model identifies how and where the repercussions of change will occur.

CONCLUSIONS

The development of the conceptual model is the first step in developing a computer simulation model. It will guide identification of the kinds of variables, measures, and sequencing necessary to stimulate human performance in the intelligence production process.

APPENDIX B

FUNCTIONAL DECOMPOSITION AND ERROR FRAMEWORK

FUNCTIONAL DECOMPOSITION AND ERROR FRAMEWORK

INTRODUCTION

This appendix describes a behavioral framework for capturing intelligence production performance in a computer simulation environment. It contains two parts: an error framework and a functional decomposition. The first is the basis for the algorithm and the other is the basis for structure for the computer model.

We previously described (see Appendix A) the conceptual model for developing a computer simulation model of how MI produces intelligence. The computer model will allow us to predict and evaluate how deficiencies in using information impact intelligence production. Modeling allows us to systematically determine the effects of variables on organizational performance in a timely and cost-effective manner. However, before modeling efforts were undertaken, we needed to determine what behaviors to simulate and to identify the parameters influencing those behaviors.

BACKGROUND

The MI system consists of individuals and teams at various organizational levels using raw or processed data to *produce* descriptive and predictive intelligence. The primary function of MI, regardless of the organizational level or structure, is to transform information from one state to another. We have conceptualized the MI system as an input-processing-output network model (see Figure B-1). Data that first enter the system are transformed at Node "A" into a different information state, "delta a." That information is passed to other nodes (functions) where it is transformed by those functions. The process continues until the final intelligence product "delta g" is produced. The path the information takes as it flows through the system depends upon the products or intelligence output required of the system.

The measures of performance within the intelligence production paradigm are the assessment of the output, either "deltas" or the transformations within the functions. The latter is a measure of the efficiency in accomplishing the transformation (e.g., how quickly tasks are performed). The former is a measure of effectiveness of the transformation (e.g., how well the intelligence meets user requirements). The Intelligence Production Performance Model (IPPM) focuses on effectiveness—the utility of information to its user. The user may be either the receiver of the final intelligence product or a functional node that transforms the data.

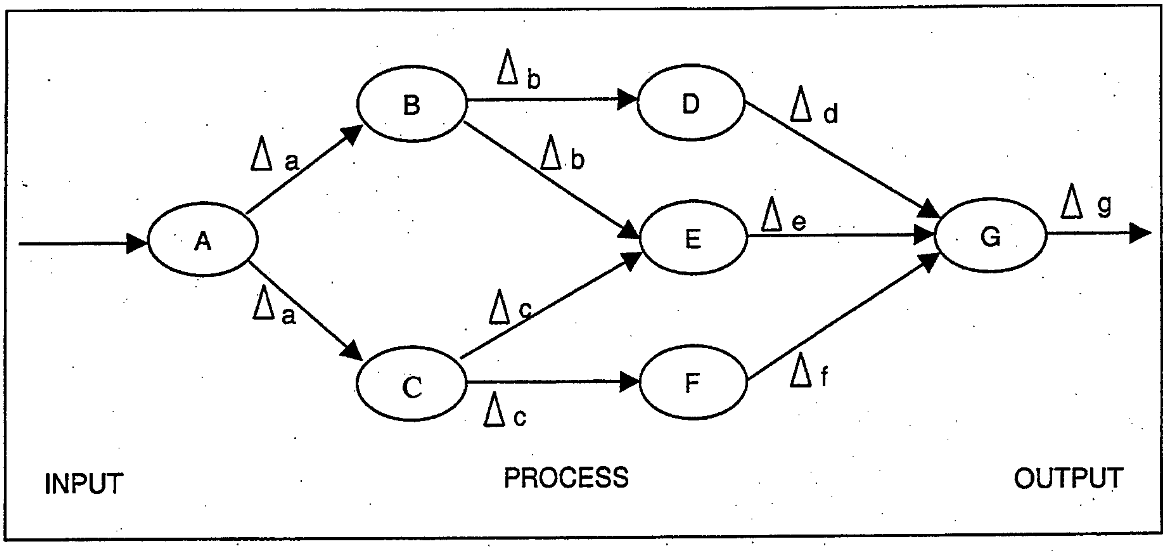

Figure B-1. The MI production conceptual model.

To evaluate performance within a computer environment, the behaviors necessary to transform information must be modeled. Most computer models of human performance employ either a task analytical or cognitive modeling approach. Neither is readily adaptable to an organizational information processing computer mode. Cognitive models are used primarily in theoretical research about how humans solve problems, and these models focus on individuals rather than organizations. Task analytical models are used to examine single or groups of people. Also, they tend to focus on task activities rather than information processes (e.g., time to complete tasks rather than quality of information transformation). Because of the limitations of existing human performance models, it was decided to pursue an error approach for modeling intelligence production performance. An assumption was made that errors made in transforming information affect the quality of the output.

Before a computer performance model based upon human error could be developed, an error framework had to be developed. It required developing an error taxonomy for intelligence production and identifying the conditions that caused errors. The framework also required a way to assess the impact(s) of error(s) on intelligence production performance.

DEVELOPMENT OF THE ERROR FRAMEWORK

Previous research (Burnstein, Fichtl, Landee-Thompson, & Thompson, 1990) had users of intelligence specify their information requirements. Using measures provided, they established

standards that identified when intelligence was considered acceptable or not acceptable. User assessments of intelligence provided the data for identifying information deficiencies in intelligence production. A fault analysis was then used to determine the source and cause of the deficient intelligence. This research led to the belief that any deficiency in intelligence came from either of two sources or their combination. Those were human "errors" in the production process and inadequate information used in producing the intelligence.

Backward chaining (see Figure B-2) was used to develop the error framework. The first step was to establish what errors would result in deficient information output to users and then determine the variables that occasioned the errors.

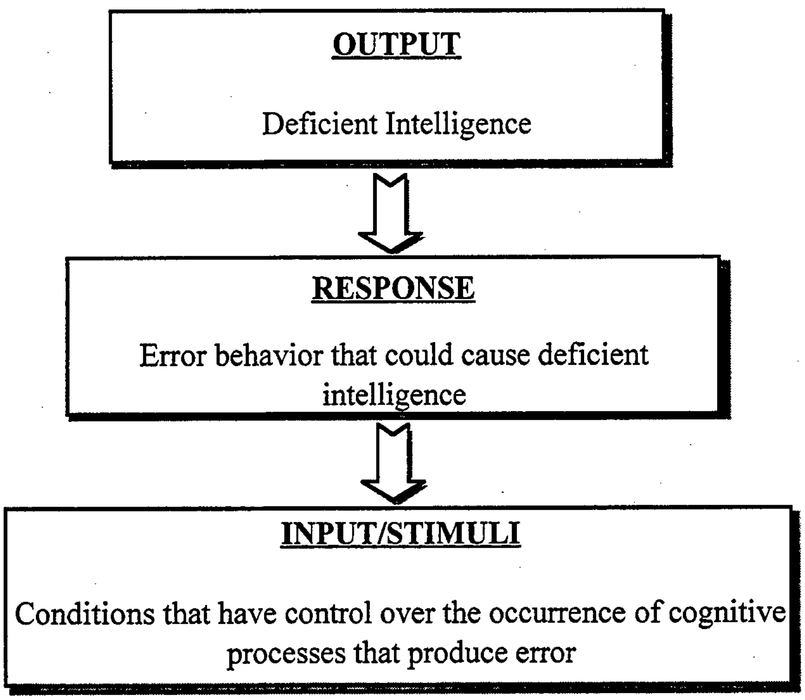

Figure B-2. Backward chaining logic used to develop the error framework.

Defining Error

Most authors agree that error is difficult to define. Rasmussen (1986) proposed that human errors should be considered as "instances of human-machine or human-task mismatches." His goal was to identify the cause of errors. Later, Rasmussen (1987a) said that errors could only be defined in relation to human intentions or expectations. They depended on someone's judgment. He pointed out that errors can be caused by changes in the judgment criteria or changes in performance with respect to accepted performance. Senders and Moray (1991)

identify several different definitions of human error; all "imply a deviation from intention, expectation, or desirability." Miller and Swain (1987) accept, "as defined by Rigby (1970), human error is any member of a set of human actions that exceeds some limit of acceptability. It is an out-of-tolerance action, where the limits of acceptable performance are defined by the system." Rouse (1990) does not define errors but states, "Errors in themselves are not particularly ***troublesome.*** The real problem is consequences. If undesirable consequences could be avoided, human errors could become, to a great extent, a non-problem."

Woods (1989) also brushes aside error definitions to stress the need for error modeling to address error causation, detection, and correction. Michon, Smiley, and Aasman (1990) define error as "simply any difference between the set-points and the actual state of a system at a given time." For them, error is "only those consequences of behavior that actually increase the distance between the present state and goal state eventually to be reached." Finally, Reason (1990) regards error as a generic term for all occasions when activities fail to accomplish the intended outcome.

Defining error is more philosophical than practical. Whatever the definition, error consistently involves a discrepancy. However, authors may not be distinguishing behavior from the consequences of behavior. The discrepancy may be between actual behavior and expected behavior or between the actual consequences and expected consequences.

The error definitions accepted for the error framework are more in line with Rouse (1990) and Woods (1989). It is important to address the problem of error. Thus, error is a behavior, not a consequence of behavior. In the conceptual model, consequences are the outputs (deltas) that result from processes involving human behaviors, including errors. The impact of the error behavior is to make the output deficient in some way (effectiveness) or cause additional demands in later functions. This enables us to incorporate value added. (Error makes value added negative.)

The error framework depicted in Figure B-2 is the antecedent conditions-behaviors-consequences paradigm (ABC). In keeping with the definition, errors represent behavior. These errors result in deficient outcomes—in our domain, deficiencies in intelligence.

Error Taxonomy

Several cognitive loaded frameworks and taxonomies of human error have been developed (Altman, 1966; Navarro, 1989; Rasmussen, 1980, 1982, 1987a, 1987b; Reason, 1987a, 1987b,

1987d; & Rouse, 1990). In addition, error types for specific cognitive tasks such as making judgments (Brehmer, 1987), abductive reasoning (Hartley, Coombs, & Dietrich, 1988), and planning (Reason, 1987c) have been suggested.

Norman (1981) classified errors based on their source. Examples are errors in the formation of the intention, from faulty activation of schema and from faulty triggering of active schema. It was difficult to use this classification because of the behavioral direction of our error framework. Rasmussen (1982) presents a multifaceted taxonomy. However, a difficulty with this taxonomy is that it mixed errors as behaviors with errors in triggering conditions.

Senders and Moray (1991) identify four methods of classifying errors:

1. Phenomenological. "These describe errors superficially with terms that refer almost directly to events as they were observed." "In applied areas, where the emphasis is on interaction with machines, classes such as recoverability, the attribution of error to either human or machine, and the nature of the consequences of errors are common."

2. Cognitive mechanisms involved. "Errors are classified according to the stages of human information processing at which they occur." For example, there are errors of perception, memory, attention, and so forth.

3. Biases or deep-rooted tendencies. For example, the confirmation bias would fall here.

4. Neurological events.

Senders and Moray (1991) also present two classifications of errors. The first by Rasmussen (1982) is also presented by Rouse (1990). The second taxonomy by Moray is a modification of Altman's (1966). The classification is based on level of behavior complexity, their mode, the type of learning involved, and psychological data from other research that helps us understand the origins of error.

Hale (1990) developed a classification in relationship to Rasmussen's (1986) skill, rule, and knowledge-based levels of behavior. His six categories, developed for the safety rule domain, were

1. Errors because a person did not know the appropriate rule to cope with the situation facing him or her and could not solve it in time (knowledge-based mistakes).

2. Errors because the inappropriate rule is selected for the circumstances (rule-based mistakes).

3. Errors in performing a routine because of failures in detailed monitoring and control mechanisms (skill-based slips).

4. Actions omitted because the relevant people did not realize that it was their responsibility to perform them.

5. Errors because the person does not switch from a routine into a higher level of behavior to cope with the exception or switches back too soon to a lower level before fully analyzing the consequences of the action chosen.

6. Actions that achieve short-term goals but result in some kind of long-range negative consequences.

Reason (1990) lists errors based on "failure mode" for the three behavior levels. Reason (1987a) provides a comprehensive framework for classifying errors. It exists as a matrix with one dimension that could be regarded as cognitive processes and the other as external variables. Within the cells are "error tendencies" that are equivalent to the errors in our list. As with other cognitive taxonomies, Reason's taxonomy is very difficult to translate to a behavioral framework.

Altman (1966) provided an error taxonomy based on phases of human performance. The phases are planning, designing and developing, producing, distributing, and operating, which is divided into information processing and decision making, and maintaining. Each phase is broken into various tasks that represent the phase. Error behaviors are identified based on the behavior level characteristic of the phases. The behavior levels are identified as

1. Sensing, detecting, identifying, coding, and classifying;

2. Chaining or rote sequencing;

3. Estimating with discrete responding and estimating with continuous responding (tracking); and

4. Problem solving.

The last example of error classification is Norman (1983). He categorized slips (read consequences) based on their source (read behavior). Unfortunately, this is also a cognitive classification that is difficult to translate.

One of the problems of classifying errors is the level of resolution necessary to capture error behavior. There may be one level of error specification, for example, "missed a step," that provides general coverage. On the other hand, there may be a particular step, at whatever level, so important to the domain that it requires specification, for example, "did not reweigh the importance of old information based on new information."

The latter example could be regarded as "missing a step." Therefore, the level of resolution for creating an error taxonomy poses a difficult problem.

The Error Taxonomy for Intelligence Production

The taxonomies and frameworks provided by Altman (1966), Rasmussen (1982), Reason (1987c), and Rouse (1990) provided guidance for the development of the error taxonomy for intelligence production. The elements and structure of these taxonomies were changed as necessary in order to be in the context of intelligence production. The resultant MI error taxonomy consisted of 54 errors classified into one general and five special categories of error. The special categories include

1. Complying with administrative requirements. Certain administrative procedures and information exist to constrain, direct, or guide behavior. Examples in the domain include unit SOPs, priority information requirements (PIRs), intelligence requirements (IRs), and various types of orders, doctrine, and supervisory and managerial idiosyncrasies.

2. Collecting information. Data are the raw material that intelligence-electronic warfare (IEW) acts upon to produce intelligence. Data include not only information collected by the INTs (general word for different types of intelligence that are collected, such as human intelligence [HUMINT], signal intelligence [SIGINT], etc.), but also information from established databases.

3. Recalling knowledge. Information required for the production of intelligence also exists within the performer as well as in databases and references.

4. Hypotheses testing and selection. Within intelligence production, assertions, assumptions, and predictions are made, based on collected and recalled information. We use the term hypothesis as a general term describing those outputs.

5. Equipment operation. When intelligence production is actually involved with equipment operation, a high level of resolution can be used to describe errors.

Since the error framework defines error as a behavior, an error can occur while a person is performing procedures or engaged in mental processes. When the error occurs in a mental process, we assume the error is based on the type of the deficiency we see in the outcome. Since the intent of the error framework is to keep the reasoning chain simple, mental errors are not considered to produce procedural errors, which in turn produce deficient performance. Within each category, errors were identified as procedural or process errors. Within that context, they are identified as errors of commission or omission. The intelligence production error taxonomy is presented in Appendix F.

Precipitating Conditions for Errors

According to the ABC paradigm, errors do not just happen. Some controlling conditions must cause them. Most studies of errors seek cognitive explanations for errors, rather than viewing errors as the result of precipitating conditions. Few studies have manipulated stimulus conditions in an attempt to determine or control the resulting types of errors. However, the stimulus control of errors has been implicit in the literature.

Norman (1981) proposed triggering conditions as a critical factor for correct performance. He also implied that these conditions might be incorporated in computer design to reduce error (Norman, 1983). Reason (1986, 1987d, 1990) discusses contextual cueing, environmental control factors, and calling conditions as specifiers for occasioning errors. Thus, Bagnara, Rizzo, and Stablum (1989) indicate there is a consensus that "human error depends on the user's knowledge organization and cognitive control and on the characteristics of the environment where the user's performance takes place." Woods (1989) stressed the need to consider how features of the domain and situation increased problem demands to produce errors. He then used the features to constrain cognitive simulation in his problem-solving model. For the most part, errors were identified, the possible cognitive contributions to the error hypothesized, and the possible stimulus conditions suggested. Baars (1980) summarizes work on actually manipulating stimulus conditions to elicit predictable speech errors, indicating that the stimulus control of specific errors is more than a conceptual issue.

While the issues of stimulus control and error behavior within a cognitive framework are typically not linked, in theory it is possible. Reason (1986) avers that "systematic forms of human error have their origins in fundamentally useful processes." Thus, it was decided that it

was feasible to develop a model in which errors would be triggered by the operational conditions of the system. By backward chaining from possible MI deficiencies, we have defined three classes of independent variables as controlling conditions for error within the intelligence domain. They are information, trigger, and information state variables.

Information Variable

The information variable is a domain-content-free description of the information sample from the environment. Military intelligence will operate on the sample of raw data and processed information at any point in time to produce intelligence. The information variable is constrained by weather, terrain, and friendly and opposing force modes of operation. The information variable identifies the errors that are systemic to the intelligence production system. If the intelligence system worked perfectly but "poor" information were being entered, deficient intelligence could result. The poor information must trigger errors that result in the deficient intelligence—thus the need for the information variable.

Trigger Variables

Trigger variables are the variables that occasion the possibility of error. In contrast with the information variable that requires an error to occur, the trigger variables cause an error to occur only when given a particular set of circumstances.

The trigger variables were selected from the many taxonomies of independent variables controlling behavior, primarily Gawron, Drury, Czaja, and Wilkins (1989). Other trigger variables were created or brought to different levels of resolution to fit the requirements of intelligence production. The variables fell into two classes: operator and operational variables (see Appendix F).

Operator variables are brought to the situation by the performer(s). They include training, experience, and personal variables. Operational variables are imposed on the performer. They include environmental, management, and task variables. The task variables are divided into those external and internal to the task.

The operator variables and the environmental, management, and external task variables are described by different levels of the variables. The internal task variables are response requirements, job aids, procedural requirements, and stimulus characteristics. They are either present or absent.

Information State

Intelligence is a domain rich in detailed information. It would be impossible to represent either richness or the extensiveness of the information content within a model. Thus, there is a need to have a domain-content-free model. The information state describes information that results from transformations occurring in the intelligence production process. It is in contrast with the information variable, the domain-content-free description input to the intelligence production system.

The information state was viewed as a vector, in which each point represented a different dimension. The value of that point represents a scale value describing the level of that dimension (see Appendix C). Since a deficient information state is the result of transformation in intelligence production, it must be capable of triggering errors. Thus, the impact of poor performance in one function must have an effect on a subsequent function.

DISCUSSION

The error framework is conceptually compatible for use in the computer model of intelligence production. If the three classes of independent variables proposed control human performance in the MI information processing system, it is possible to manipulate them to produce or eliminate error. For example, if all variables are at an optimal level, no errors will be made, but if there is a deviation from the optimal, errors will occur. The errors committed depend upon the information transformation being performed. Therefore, for each information transformation function, a set of errors can occur, contingent upon variable sets representing the real world situation.

Although the objective of the MI system is to process information, the three variable classes controlling errors can be defined independently of the cognitive processes involved. For example, training and experience can represent the cognitive content and procedural knowledge a person brings to the job. Successful past job experience is an indication of cognitive knowledge. In addition, the amount of training and the type and years of work experience are indirect measures of cognitive knowledge. The assumption is that performers use or remember what they were trained in or learned through experience. Thus, the cognitive dimensions can be captured by the proposed variables, although indirectly, without getting directly involved in specific cognitive processes where the state of the art is fuzzy and vague.

The error framework will enable us to manipulate the factors that exert control over error behavior. If we can validly model performance deficiencies, then more appropriate cognitive "prostheses" (Reason, 1987e) should be possible for reducing errors. For example, decision support systems can be used to enhance decision making by manipulating the stimulus environment of the user. In addition, these systems can direct the user to critical information, provide feedback about past and current performance, warn of errors of commission, and make tasks less difficult and time consuming without invoking cognitive impact. Reason (1986) avers that "systematic forms of human error have their origins in fundamentally useful processes." Since the same cognitive processes used in decision making are the ones that create errors, the reduction of errors in intelligence production, through the manipulation of the three variable classes, is reasonable.

In order to use the error framework as part of the computer simulation model, we needed to describe MI as a functional organization. This description enabled us to identify the variables demanded by the error framework, as well as other features of intelligence production that might be necessary for the computer model.

THE FUNCTIONAL MODEL

The functional model operationalized the conceptual model in the MI domain. The model consists of the functions required to produce intelligence and their decomposition. It was developed independent of any MI operational structure. The functional model was produced by SMEs having G-2 or S-2 experience and with reference to current MI doctrine. Psychologists in consultation with SMEs determined behavioral aspects of the decomposition.

Four major functions identified were (a) battlefield assessment, (b) collection management, (c) collection, and (d) data evaluation, analysis, and integration. Except for the collection function, the other functions were further decomposed.

Functional Flow

A general depiction of the functional flow is shown in Figure B-3. The outer ring represents the data evaluation, analysis, and integration functions, and the next ring with the dark node represents the collection function. The third ring (cross hatched) is collection management, and the innermost ring (shaded) is battlefield assessment. Arrows indicate the direction of the functional

flow. The rectangles represent organizations outside the production that receive and send information to the intelligence production functions.

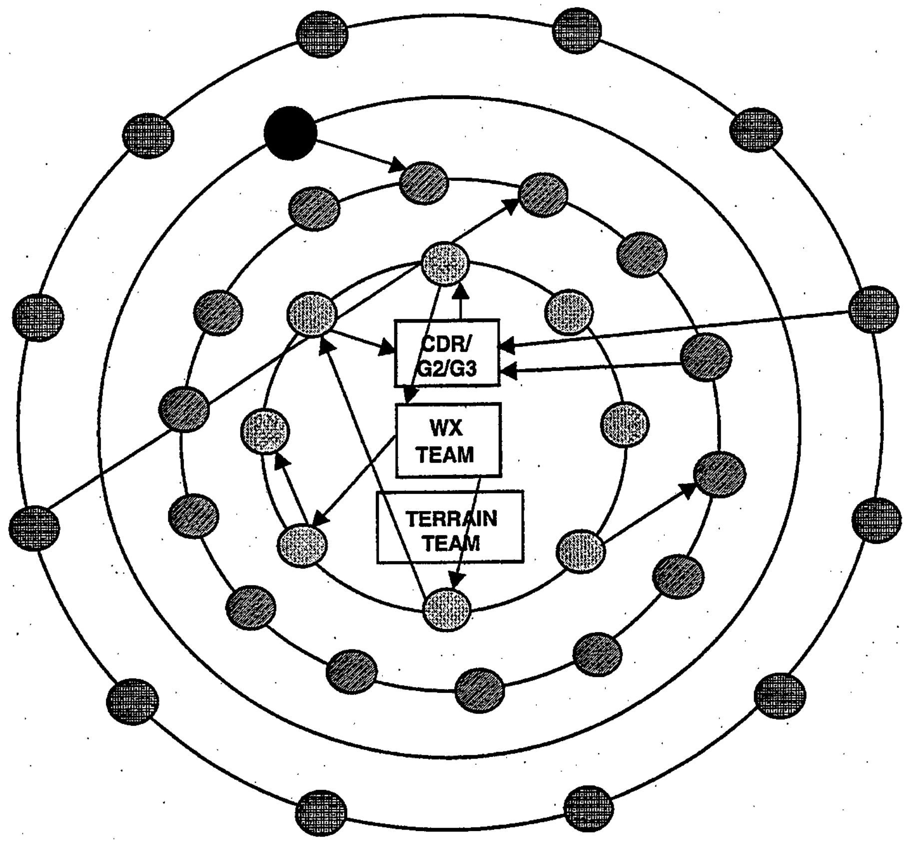

Figure B-3. General depiction of the MI functional flow.

The functional flow represents the order in which the functions are performed. According to the conceptual model, the functions are sequential, that is, a function cannot be performed unless the previous function has occurred. Although intelligence production is a continuous process, the functional model is not meant to represent processes in that manner. Although it shows that some functions may operate in parallel, the functional model shows no recycling as may occur in actual intelligence production.

Decomposition Within Nodes

An example of the decomposition for a node is shown in Figure B-4. The following is a discussion of the decomposition in relation to this example:

2.1.2.3: Given specific indicators, DETERMINE ENEMY NODES, ACTIVITIES, AND EVENTS THAT WILL PROVIDE INDICATORS for specific information requirements (SIR):

1. FUNCTION IDENTIFIERS: 2.0, 2.1, 2.1.2

2. INPUT (SOURCES):
   Specific indicators (2.1.2.2)
   ASPS database

3. OUTPUT/DESTINATION: Specific Information Requirements (SIR) to 2.2.1

4. STIMULUS AND RESPONSE VARIABLES

Visual	Verbal communication
Fine Motor	Written communication
Recall	Hard copy visual
Analyze	Soft copy visual
Integrate	Symbology
Evaluate	Graphics

5. GENERAL ERRORS: GPC1, GPC4, GPC5, GPO1, GPRC3, GPRO1

6. ELEMENTS of 2.1.2.3	7. SPECIFIC ERRORS
1. Select all enemy nodes, activities, events that will provide answer to collection task requirements.	
2. Select critical enemy nodes, activities, events that will provide most substantive indicators.	GPRC1, GPRC2, GPRO6, CPRC2, CPRC3
3. Select other enemy nodes, activities, events to support or refute substantive indicators.	CPRC2, CPRC3, HPPRC1, HPPRC2, HPPRC3, HPPRC4, HPRC1, HPPRC6, HPPRO1

8. PROCEDURAL KNOWLEDGE REQUIREMENTS: Sequential

9. DOMAIN KNOWLEDGE REQUIREMENTS
   OPFOR tactics and capabilities
   OPFOR order of battle

10. JOB AIDS
    Reference tables, charts, manuals, maps
    Templates

11. ABILITIES

Written comprehension	Inductive reasoning
Written expression	Category flexibility
Memorization	Deductive reasoning
Originality	Mathematical reasoning
Fluency of ideas	Number facility
Spatial orientation	Perceptual speed and accuracy
Visualization	Visual color discrimination

12. HUMAN-MACHINE RELATIONSHIP: Given the high cognitive demand, this is better done by a human.

13. PROBLEM-SOLVING STRATEGY: Inferencing

Figure B-4. Example decomposition of an MI function.

0. 2.1.2.3. It is the node identifier and it includes the input-process-output description of the node.

1. Functional identifiers. These are codes that identify the location of the node within the total decomposition.

2. Input sources. They are input from other nodes, as specified. The specific indicators are an information state variable. If deficient, they would occasion errors. With output or destination, they also determine the functional paths of information in the production system. Other input (e.g., all-source production system [ASPS] database) are part of the information variable. The databases are often common to several nodes.

3. Output-destination. Identifies the information product of the node and where it is sent.

4. Stimulus and response variables. These are the trigger variables that operate at this node. They were determined based on analysis of the tasks required by the node as currently performed.

5. General errors. These errors, from the error taxonomy, could occur while the processes required by the node are performed.

6. Elements. These are a further decomposition of the node into tasks. When possible, this was done to help determine the trigger variables and errors operating at a node.

7. Specific errors. When possible, errors from the taxonomy were attached to task elements. If not, the errors were classified as general errors.

8. Procedural knowledge requirements. This contains two items: the first is the procedural requirements from the trigger variable list (e.g., sequential) and the second is the procedural knowledge required to perform a function. This information helped to determine the possible errors for the node. While not intended to be part of the computer model, these requirements can assist in the analysis of the model's predictions and evaluations. For example, if an error could be tracked back to a procedural deficiency, we could be addressing a training or design problem. Thus, these requirements may help in diagnosing and identifying potential remedies.

9. Domain knowledge requirements. This is knowledge of the domain of MI required by the node. It is of value for diagnosing and identifying potential remedies.

10. Job aids. As the job is currently performed, the job aids used are listed. These also relate to trigger variables.

11. Abilities. These are the human ability demands, based on the tasks required to perform the transformations at the node. While not intended to be in the model, the abilities reflect possible soldier selection criteria. Thus, they can assist in analyzing the model outcome for diagnosing or identifying potential remedies. The abilities were determined by using the Job Comparison and Analysis Tool (JCAT) on each node (Muckler, Seven, & Akman, 1990).

12. Human-machine relationship. An estimate was made of how to distribute the labor necessary to perform at the node.

13. Problem-solving strategy. This was not a part of the original decomposition. It was based on later research (Warner & Burnstein, 1996). It identifies the kind of problem represented by the node. This leads to the kind of design template needed to aid in redesigning or controlling behavior and variables at the node.

Function Characteristics

In addition to decomposition, the information output from each node was identified. Since we were concerned with a domain-content-free model, the output was identified as information state. Information state was described by seven dimensions (see Appendix F), although not all the dimensions were appropriate at each node. The result was an "exemplar" information state vector for each node. The exemplar described the expected output of the node by the dimensions the content output would have and the level of the dimensions, given a perfect information variable and errorless performance during the processing.

Finally, for each node, the impact of errors on the exemplar information state vector was determined. That is, for each error that could occur at a node, we determined its effect on each dimension within the exemplar information state vector. How errors affect the exemplar vector is discussed further in Appendix C. In addition, the resulting matrices for the functional characteristics form the supporting data tables for the computer mode.

The completion of the error framework and functional model provided the basis for development of a logical model. The logical model would provide the guidance for development of the computer simulation model.

APPENDIX C

THE LOGICAL MODEL

THE LOGICAL MODEL

BACKGROUND

This appendix presents the logical model of intelligence production. The logical model provided the guidance for the development of the computer simulation model. It integrated the functional decomposition and error framework within the constraints and assumptions defined for the computer model.

Recent advances in technology have caused MI to change their way of doing business. In addition, the austere budget conditions have made it crucial to understand the impacts of those changes to ensure that intelligence users do not get a degraded product. The most cost-effective way to assess those impacts is through simulation and modeling.

We first developed a conceptual model of intelligence production (see Appendix A) that conceptualized the MI system as an input-process-output network model. The conceptual model was expanded into a functional model by identifying the functions in the MI system. The functions were then decomposed to determine the variables and human performance requirements within the functions. An error framework (see Appendix B) that would serve as the basis for an algorithm within the computer model was also developed. The error framework included a taxonomy of errors for intelligence production, the variables that occasioned the errors, and a theory of how errors operated.

The problem of how to model an information production system was bound by making some basic assumptions and identifying some constraints:

1. Previous research that assessed MI effectiveness provided organizational performance criteria. Since that gave us a solid base from which to proceed, we constrained the modeling effort to measuring effectiveness. If we could produce a computer model for intelligence production effectiveness, efficiency could be added later (conceptually), if necessary.

2. Most computer models of human performance are high resolution and measure the efficiency of performance. In fact, they are often very behavioral as well as performance oriented. The results of these modeling efforts often have little generality outside the situation being simulated. We assumed that for the model's results to be generalizable, that is, not situationally bound, the model needed to be low resolution. In addition, we assumed that the model should be free of domain content imposed by situational context or intelligence operations.

3. Initially, the computer model was limited to a single intelligence function. Since intelligence production covers many aspects of behavior, we decided to concentrate on a function that had a high human cognitive requirement. Because of the constraints for a high cognitive-loaded function and a low-resolution model, we decided to model impact of errors on performance.

LOGICAL MODEL

The error framework and the functional decomposition needed to be integrated before programming of the IPPM could begin. The logical model represented that integration, given the assumptions and constraints of the modeling effort. It describes what the model must do, the information required to do it, and how the information is used. It was developed to try to ensure that we had identified all the events, information, and rules that were necessary to produce the IPPM. It was developed independent of user interfaces and database structures.

Figure C-1 indicates that change impacts intelligence production by altering the trigger variables and information variable. The result is a new error simulation that affects the intelligence being sent to the consumer. Thus, any change that can be translated into trigger variables or the information variables can be assessed by the model.

Figure C-2 conceptualizes the identification of the information variable and trigger variables. A scenario sets the occasion for describing the battlefield situation. The battlefield situation includes the weather, terrain, the enemy and friendly forces, and their modes of operation. The information requirements and MI operations are then determined. In addition, constraints on what information can be collected are identified. From these elements, the information variable and the trigger variables are established.

Figure C-3 conceptualizes how error performance should be simulated. The information and trigger variables determine the errors that are possible during a model run. An error algorithm acts upon these errors and determines which errors will be triggered. When errors are triggered, they degrade what would have been the exemplar output of the function (measured in terms of the function receiving that node). This degradation serves to occasion errors in the next (receiving) function. These errors, plus the ones previously identified by the information variables and trigger variables, form a new error set for the next function, and the process continues.

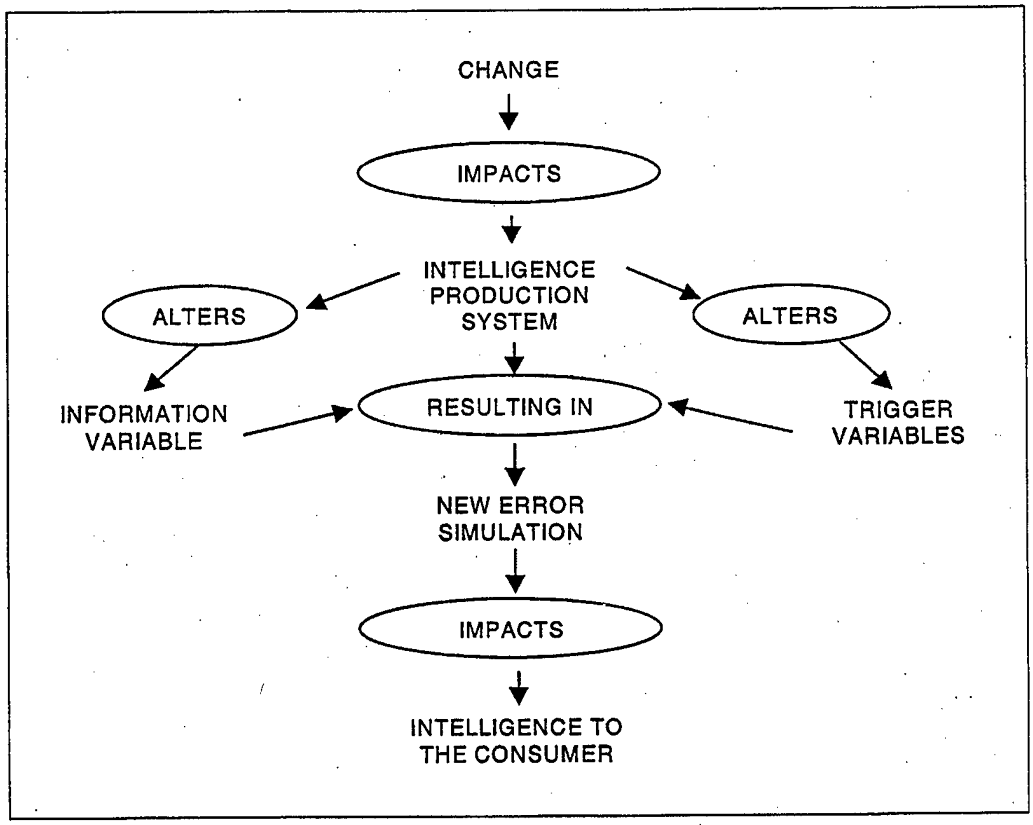

Figure C-1. How change affects the intelligence production system.

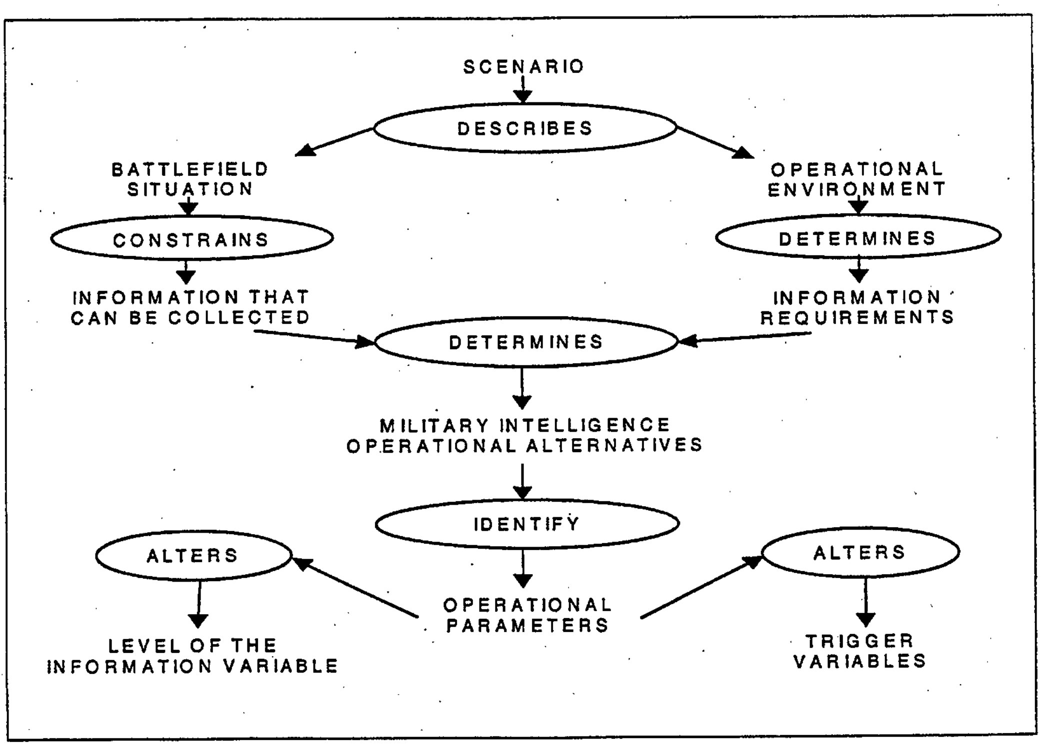

Figure C-2. The identification of the information variable and trigger variables.

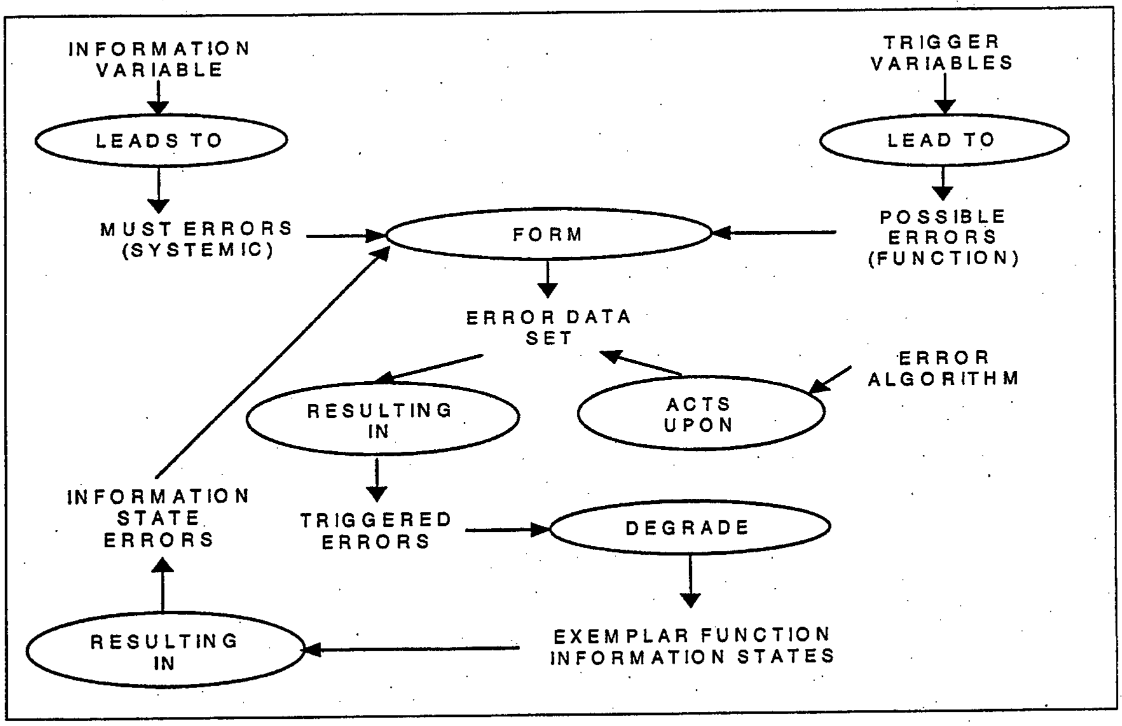

Figure C-3. How error performance should be simulated.

Information Variable

The information variable describes the information being acted upon by the intelligence production system to produce intelligence. The information variables are the means of operationally defining a sample of information from the environment as a non-domain-content description. Manipulation of the level of the information variable represents information changes in the intelligence production system.[1]

Operationally, the information variable is described as the "information" dimension completeness. Completeness is the degree that the sample of information from the battlefield contains the information necessary for the intelligence production system to satisfy the intelligence producer (internal) and the intelligence consumer (external). Five levels of completeness have been proposed.

The information variable is derived from the battlefield situation and the operational environment. The battlefield situation includes the terrain, weather, or other situational factors

[1]The asset collection suite in the IPM, which was described in the main document, replaced this simple variable by simulating collection variables. Appendix E describes the collection modeling approach and how it instantiates the information variable.

that determine what can be sampled from the battlefield. The operational environment includes the friendly force and opposing force modes of operation that determine what can be sampled from the battlefield situation. The battlefield situation and operational environment constrain the information that can be gathered and therefore determine the completeness level of the information variable.

Interpretation of the battlefield situation and specification of the operational environment are based on the predefined operational situation and knowledge bases such as doctrine, lessons learned, and subject matter expertise. This knowledge is not expected to be part of the computer model. It would come from a user having specific questions to be answered. As a result, there are no rules or guidance for determining how to set the level of the information variable.

Operational Parameters

The operational parameters were believed necessary for specifying the operational environment being simulated. SMEs identified the operational parameters (see Appendix F) and determined the trigger variables or level of trigger variable that each operational parameter would set. Six classes of operational parameters were identified: echelon, mission, physical combat environment, psychological environment, enemy technological capabilities, and friendly technological capabilities. Appendix F contains all model terms and definitions.

An example of the relationship of different operational parameters (OPs) to trigger variables is the operational parameter freedom of action that has three levels. It can set the trigger variables' formal controls and management style. Level 2 of this OP would set the formal controls' trigger variable at Level 2.

It is evident that a description of the operational environment could require several OPs that relate to the same trigger variable but at different levels. We addressed this problem by identifying OP precedence. For example, if all OPs were required, then battlefield operations would be the determining parameter for trigger variables formal controls, temporal constraints, and management style. In addition, if the description of the operational environment were so limited that none set a trigger variable level, then a trigger variables default was required.

Identifying the Intelligence Production Functions

The purpose of identifying the intelligence functions is to limit the functions to be simulated and the variables and conditions for those functions. The selection of the function(s)

limit(s) the scope of the simulation. The selection determines the beginning and end point for the simulation. Each function in the simulation becomes a defining structure of limiting variables and conditions within a simulation. The functions available for the simulation were derived by functionally decomposing a model of the intelligence production system (see Appendix B). The functions identified for a simulation depend on the questions to be answered. Thus, the functions simulated are user dependent and not inherent to the logical model.

When functions are selected for simulation, the following information is required for each:

1. The exemplar Information State vectors. The Information State is a non-domain-content description of the output of a function. It describes the output that would occur if the function received optimum information and operated without error. Information State is multi-dimensional. It is a vector in which each element represents a scale value describing that dimension (see Appendix F).

2. The trigger variables. Trigger variables are operator and operational variables (see Appendix F) that can cause errors to occur. These are necessary for defining the baseline performance and are the means to represent changes in the production system. Each function has a predefined set of trigger variables, based on current MI operational procedures. Therefore, when any function is identified, the predefined trigger variables operating within that function must be identified. The OPs determined by the battlefield situation also reduce the trigger variables within each function.

Establishing Error Conditions

The purpose of establishing the error conditions is to define the errors that will operate in the simulation.

Identification of Must Errors

A "must" error represents the error equivalents of the information variable. They were identified by determining errors that had a direct relationship to the different levels of the information variable.

A must error is required because the model must evaluate the effects of manipulating the sample information from the battlefield situation, given that the intelligence production system is operating "perfectly." Since the algorithm for the model is the error

framework, the information variable must cause errors throughout the system. In addition, a must error cannot be corrected during the simulation.

Identification of Possible Errors

According to the error framework, the operator and operational variables (trigger variables) set the occasion for errors. This means that an error may be possible, but it need not occur.

To determine the "possible" errors, the potential for errors for a function must be overlaid with the trigger variables operating at that function. To facilitate the matching required by the logical model, errors were associated with trigger variables independently of any function. The combination of the error-trigger variable, the error-function association, and the trigger variable-function association identifies the possible errors for each function in the simulation.

Determination of Trigger Levels for Errors

Since possible errors do not have to occur, an algorithm is necessary to cause them to become operational during the simulation.

In the development of the trigger variable-error association, each error was paired with each level of trigger variable. For each pair, a confidence level was estimated. The confidence level represented how confident we were that the error would occur, given the variable or level of the variable. Five confidence levels of confidence factors (CFs) were established: the error is inevitable (100), it was expected to occur more often than not (75), it was just as likely to occur as not (50), it could occur but was not likely (25), and it could not occur (0).

There were two kinds of data that could be used for the triggering algorithms. One was the number of different trigger variables that were operating to occasion an error. The other was the distribution of the confidence levels for the errors. For example, an error could be occasioned by 12 different trigger variables. The confidence level distribution might be one at 75, six at 50, two at 25, and three at 0.

The actual algorithm should consist of a set of rules and each must be tested for the errors to be triggered. While a "strawman" hypothesis for levels and rules would be developed to facilitate computer programming, the final algorithm will be determined through sensitivity testing (see Appendix D).

Determining Error Impacts

The effect of errors is to change the values of the dimensions in the exemplar information state vectors. The impact of any error depends on the function and the information state dimension operated on by the function. When an error is triggered at a function, it has a predetermined effect on the dimensions in the information state vector.

In the development of the error-function association, SMEs determined which dimensions of the vector each error would act upon. In addition, they determined the impact of the error. For example, a function operates on relevance and the exemplar point is "relevant." The triggering of one error could cause that dimension to be changed to "wrong relevance," another error could change it to "no relevance," or another error could have no effect.

Since multiple errors can be triggered and errors can have different impacts, rules must determine the impact to use. Two possible options are available. First, no matter how many errors are triggered or what the different impacts are, always select the worst impact. The second option is based on predominance. In this case, ties are possible and a rule is necessary. For example, use the predominant impact unless there are ties. If there are ties, use the predominant with the best case. Thus, if for example, five errors were triggered and three change relevance to "wrong" and two changed it to "limited," "wrong" would be the impact selected. If, on the other hand, two errors resulted in "no change," two in "no relevance," and the other "limited relevance," then "no change" would be accepted.

Once an error is triggered and the appropriate impact identified, the exemplar information state vector must be degraded to reflect that impact.

Adjustment of Errors to Represent Information State Vector Degradation

According to the conceptual model, the degraded information state output from a function has an impact on the later function(s). The error algorithm requires the degraded information states to be converted into errors or error equivalents. These errors become part of the subsequent error set that can occasion errors in the next function. With one exception, a degraded information state dimension results in occasioned errors.

As with the informational variable, only the information state completeness has error equivalents. A degraded completeness dimension can have two levels: "some" or "none."

Depending on the level, the appropriate must errors are triggered when they occur in subsequent functions of that simulation.

To determine possible errors, the levels of degradation were treated as trigger variables. SMEs determined what possible errors would occur at each function for the levels of degradation. The error impacts and confidence levels remain the same.

Once degraded Information State determined errors have been identified for a function, they are combined with the previously identified must and possible errors. The error pool is then acted upon by the error algorithm to continue running the model.

SUMMARY

The logical model defined the data requirements for the computer simulation model as (a) function-defined variables, errors, and their relationships, (b) exemplar state vectors for each function, (c) error impacts for each error, for each dimension in the exemplar state for each function, and (d) the translation of degraded information state dimensions into errors.

The logical model also identified the need for two algorithms: one to determine which errors would be triggered at each function, and one to determine the impact on the information state when different errors had different impacts on the same dimension.

APPENDIX D

VERIFICATION AND SENSITIVITY TESTING

VERIFICATION AND SENSITIVITY TESTING

INTRODUCTION

Recent advances in technology have changed the way that MI does business. At the same time, the changing military is operating under austere budgetary conditions, undergoing realignment, and facing significant changes in doctrine. It is critical that the impact of these changes on intelligence production be understood. The most cost-effective manner of assessing these impacts is through modeling and simulation. The IPPM was developed to assess the impact of change on intelligence.

This appendix describes the verification and sensitivity testing of the IPPM. Testing, as an important part of model development, verifies that the computer model functions as it was intended and determines if changes in model parameters result in appropriate changes in output (sensitivity). Test results lead to a refinement in underlying assumptions, processing rules, and algorithms. Testing activity, as a whole, increases the credibility and reliability of the model.

Figure D-1 depicts the feedback that results from testing and also represents the framework for the IPPM: Conceptual, Functional, and Logical Models, and error framework, all of which have been instantiated in the IPPM computer model.

As depicted in Figure D-1, the Conceptual Model (see Appendix A) was first developed and guided the development of the subsequent entities. Given the assumptions and constraints of the Conceptual Model, the Functional Model and error framework (see Appendix B) were developed and were integrated in the Logical Model (see Appendix C). The Logical Model, in turn, provided guidance for the IPPM computer model. The feedback loop from verification and sensitivity testing indicates that results can lead to modification of these entities (except for the Conceptual Model).

VERIFICATION TESTING

The purpose of verification testing was to determine if the computer model operated correctly and if it conformed to the Logical Model. Thorough testing requires repeated model runs with varied data; verification was run concurrently with sensitivity testing to maximize the time expended on computer runs. However, test questions were formulated and results inspected independently, as they are reported herein. There were three questions of interest during verification testing:

1. Does the computer model work as designed?

2. Were the data tables constructed correctly?

    a. Are the specified error pools correct?
    b. Are the subsequent errors correct?

3. Was the error framework correct?

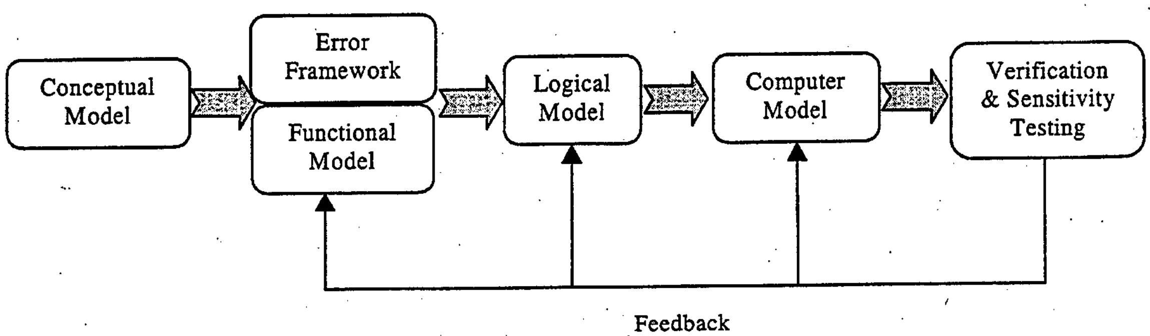

Figure D-1. Verification and sensitivity testing feedback loop.

Does the Computer Model Work as Designed?

In order to determine if the computer model worked as it was designed, we used traditional software testing methods, including unit and regression testing. Unit testing involved walking through each separate module of the software to ensure that the correct output followed from the input. As a problem was identified, it was resolved and again tested. Once unit testing was complete, regression testing began. Regression testing involved testing the computer model in its entirety, mapping input to output to ensure that no problems arose from the introduction of the new or changed modules.

Results of Unit and Regression Testing

Throughout verification testing, software changes were made in accordance with results—resulting in continuous unit and regression testing, until software changes ceased. Since extensive debugging preceded the test phase, few changes were necessitated by incorrect output. However, testing revealed that the computer model took a very long time to process. Two software changes resulted: code optimization and database restructure. Code optimization involved code changes that made the computer model run more efficiently; this included redesign of the primary module used for reporting and the distribution-building module. Database

restructure involved the combination of like, static tables (tables not altered during run time), thereby reducing their numbers and the run time associated with input-output (IO) operations. Restructuring also made software maintenance easier, more efficient, and reliable.

Were the Data Tables Constructed Correctly?

The approach to testing data table construction and the appropriateness of the error framework was similar, although the inspection of their output differed. We begin our discussion with the question of data table construction. The analysis of data table construction results focused on the error pools that were specified by the Functional Model for each node and on subsequent errors that result from degraded information state dimensions (ISDs).

Are the Specified Error Pools Correct?

Our interest in errors resulted primarily from a rather liberal approach to error specification. This approach yielded a pool of all *possible* errors being specified at each node, without regard to likelihood. The test question regarding error pools then focused on the number and type of errors being triggered at each node, given the nodal function and the trigger variable setup. Errors were inspected from simulations in which all nodes were run independently and only one operator variable (training) was manipulated. Nodes were run independently so that the impact of the trigger variable could be studied in its purest form—without compounded effects from degraded information states. Training was selected for manipulation because it has the most straightforward and predictable results (all other trigger variables remained at their default levels). Figure D-2 depicts the generalized test case for error specification.

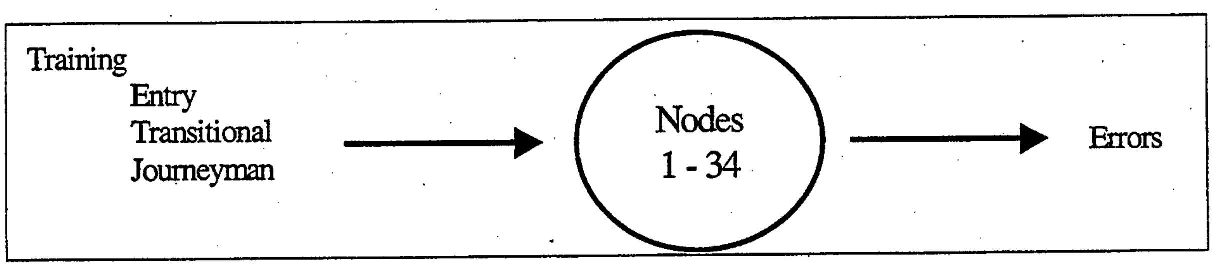

Figure D-2. Generalized test case for error specification testing.

The criterion for determining the appropriateness of triggered errors was face validity. It was critical that the output reasonably reflect that which would be expected in the "real world," based on the situation described in the computer model setup. Psychologists and

MI SMEs, familiar with the Conceptual Model and error framework, analyzed the results to determine the face validity of the output.

Results of Error Specification Testing

Results were inspected for each node, for each level of training (journeyman is the default training level, and thus, testing of it was not necessary). Table D-1 provides an example of the error specification test output from Node 1.1.1, Determine Weather Information Requirements. All trigger variables were at their default levels except for training, which was set at "entry." The algorithms under which these results were generated were "predominant" and "or." The errors in the left-hand column form the pool of possible errors specified in the Functional Model. The middle column indicates with an "X" those that were triggered by entry level training. The determination of weather information requirements involves the formulation of a request for weather data concerning a geographic area during the time frame associated with an operation.

Table D-1

Possible and Probable Errors Triggered by Entry Level Training at Node 1.1.1, "Determine Weather Information Requirements"

	Possible error pool	Errors triggered	Probable error pool
APO1	Did not consider existing administrative constraints, direction, or guidance		
APO2	Did not consider all the necessary administrative constraints, direction, or guidance	x	
APRC1	Misinterpreted the administrative constraints, direction or guidance	x	
CPRC1	Recalled more information than was necessary to perform the task		CPRC1
CPRC2	Recalled inappropriate information		
CPRC3	Did not recall all the information required to perform the task	x	CPRC3
GPC1	Perform steps incorrectly	x	GPC1
GPO1	Omit a required step		GPO1
GPRC1	Misinterpreted the information being acted upon	x	
GPRC2	Gave information more importance than necessary	x	
GPR01	Only used part of the information that is required to perform the step	x	GPRO1

MI SME analysis indicated that, given the node function, an analyst would be unlikely to commit errors associated with administrative information and guidance. Furthermore,

the likelihood of recalling inappropriate information was negligible, as was the likelihood of misinterpreting information or giving it more importance. The more probable errors associated with this node are depicted in the far right column. Considering only these probable errors, those actually triggered (committed) by an entry-level analyst are reduced from seven to three. The probable errors and the resulting list of those triggered are more consistent with what could be expected in the "real world," increasing face validity to an acceptable level.

All nodes were analyzed in this way. Further results were available for each node by virtue of concurrent sensitivity testing. Algorithm testing (described later, in sensitivity testing) provided additional test results that were consistent with those depicted in Table D-1. The overall result of error specification testing was to reduce the pool of errors at each node from possible errors to probable errors that tended to reduce the number of errors triggered.

Are Subsequent Errors Correct?

Within the computer model, the effect of triggered errors, at a given node, is to degrade the ISDs acted upon by the node. Following the Conceptual Model, this degraded ISD, in turn, impacts the next, or receiving, node. The error algorithm requires the degraded information to be in the form of errors; these errors are referred to as "subsequent errors." Originally, the Logical Model specified that the impact of subsequent errors would be node specific and similar to trigger variables. For each node, each level of the (relevant) ISDs was associated with the error pool, on the basis of what errors would possibly occur. Confidence factors were specified for each ISD level and error combination, and the result was a subsequent error data table for each node that contained confidence factors uniquely associated with the ISD level, the error, and the node.

This approach to the degraded ISDs differs from that of the information variable (IV). Each level of the degraded sample of battlefield information has defined error equivalents. The application of these error equivalents is consistent across nodes and is thus applied globally. This logical difference between ISDs and the IV was identified in the Conceptual Model and planned to be a subject of testing once the computer model was completed.

The subsequent errors produced by the subsequent error tables were verified in the same way that the error pools were verified. That is, the number and type of subsequent errors were inspected. Again, the criteria were face validity, as judged by the MI SMEs and behavioral scientists. In order to test the validity of the subsequent errors occasioned by

degraded ISDs, it was necessary to run the tests on multiple nodes or clusters. Figure D-3 shows a sample test case from the simulation run for the purpose of inspecting subsequent errors.

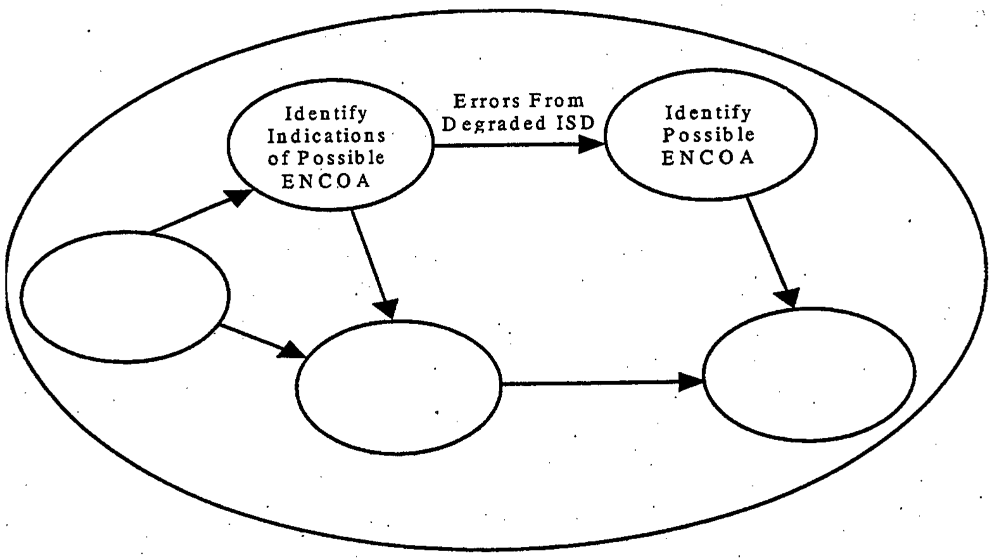

Figure D-3. Notional test case for testing treatment of the ISDs.

The tested cluster reflected in the figure, consisted of nine nodes that simulate the war gaming of possible enemy courses of action (ENCOAs) to determine the most probable one. Figure D-3 identifies two nodes from that cluster, Node 4.2.3.4, Identify Existing Indicators of Possible ENCOAs, and Node 4.2.3.5, Identify Possible ENCOAs. The purpose of the test was to examine the number and type of subsequent errors that were passed from the identification of existing indicator nodes to the identification of possible ENCOA nodes and to determine their face validity. All nodes were inspected in this manner.

Results of Subsequent Error Testing

Applying face validity criteria, the results of subsequent error testing generally indicated that too many subsequent errors were being occasioned. Not only did the subsequent error distribution contain too many errors, but also the number of each type was too great. Thus, treatment of degraded ISDs via the subsequent error table resulted in too many errors and too many error types.

The alternative to treating degraded ISDs as node-specific variables, with reference to the subsequent error data tables, is to assign error equivalents to the degraded ISDs and apply these error equivalents globally to all nodes. This approach brings degraded ISDs in

line with the IV, where errors have a direct relationship to the levels of the IV and are applied globally across all nodes. To derive error equivalents for degraded ISDs, SMEs associated each (degraded) level of the ISDs and the errors to which they were equivalent and assigned the appropriate confidence factor (CF).

To illustrate these results with the sample test case from Figure D-3, Table D-2 contains the error distribution that resulted from subsequent errors attributable to the degraded ISD. The node depicted in Table D-2 is 4.2.3.5, Identify Possible ENCOAs. The subsequent errors were passed from Node 4.2.3.4, Identify Existing Indicators of Possible ENCOAs. Trigger variables were changed for this test to simulate an organization that had poor or nonexistent procedural guides, no automated tools, poor leadership, and few senior analysts. This simulation can be thought of as representing a worst case, yet realistic, scenario. The algorithm combination used was "predominant case" and "and."

The left side of the table represents the subsequent error distribution resulting only from the degraded ISD (that will later be added to the trigger variable and IV error distributions). The columns headed "100," "75," and "50" represent the CF levels, and the numbers in those columns represent the error count for the given level. Table D-3 provides the degraded levels of the relevant ISDs (shown in bold) for Node 4.3.2.3, Identifying Existing Indicators of ENCOAs, which feed the node being discussed (all levels of the relevant ISDs are shown for comparison purposes—perishability is not relevant at this node). The impact of degraded ISDs is the subsequent errors shown in Table D-2; 19 different errors happened 42 times (across CFs). SME analysis indicated that even the poor conditions simulated in this test (realistically reflective of more extreme conditions) could not be expected to produce errors of this type and in these amounts.

The right side of Table D-2 represents the error equivalent instantiated to increase the face validity of results. The error equivalents for the degraded ISDs (from Table D-3) are shown with their resulting distribution. The reduction of error types (from 19 to 5) and number of errors occasioned (from 42 to 7) is clear and in alignment with the real world correlation of the simulation.

This test case and the results illustrate all subsequent error testing, both for other nodes within the cluster described and for other clusters of nodes run during other simulations. In all cases, the node-specific trigger variable treatment of the degraded ISDs resulted in an unrealistic profusion of errors. The error equivalent approach, on the other hand, occasions types of errors that are more in alignment with what one could expect in the "real

world," as are their numbers. An ancillary benefit to this approach was the reduction of 34 very large subsequent error data tables (one per node) to one fairly small error equivalent table, not only reducing processing time but also enhancing the ease and accuracy of data table maintenance.

Table D-2

Test Results From Testing the Appropriateness of Subsequent Errors for Node 4.2.3.5, "Identify Possible ENCOAs"

	Subsequent errors	100	75	50	Equivalent errors	100	75	50
CPC1	Collected more data than was required to perform the task	0	0	1				
CPRC2	Recalled inappropriate information	0	1	1				
CPRC3	Did not recall all the required information to perform the task	0	1	1				
GPC1	Perform the steps incorrectly	1	2	1				
GPC5	Perform the step before there is enough information to justify doing so	0	2	0				
GPC6	Perform a step too late	0	1	1				
GPO2	Stop the procedure before completing all the steps	0	1	1				
GPRC1	Misinterpret the information being acted upon	0	2	1	GPRC1	0	0	2
GPRC2	Gave information more importance than necessary	1	2	0				
GPRO1	Only use part of the information that is required to perform the step	0	1	2	GPRO1	0	0	1
GPRO3	Did not integrate new information with existing information	0	1	0				
GPRO5	Did not build models of events from a mix of hypotheses and facts	0	0	1				
HPPRC1	Used incorrect information to verify or refute predications	0	2	1	HPPRC1	0	1	0
HPPRC2	Rejected hypotheses without fully testing the predictions	0	2	1				
HPPRC3	Accepted hypotheses without fully testing the predictions	0	2	1				
HPPRC4	Tested hypotheses to a point of diminishing returns	0	1	1				
HPPRC5	Selected a hypotheses having no relationship to current or future possible friendly force or opposing force operations	0	1	0				
HPPRC6	The hypotheses selected were not supported by the existing information	0	1	0	HPPRC6	0	0	1
HPRC1	Misinterpreted the information used to verify or refute the hypotheses	0	1	2	HPRC1	0	0	2

Table D-3

Degraded ISDs (in bold) for Node 4.2.3.4, "Identify Existing Indicators of ENCOAs"

Relevance	Specificity	Completeness	Perishability	Validity	Accuracy	Redundancy
Relevant	Precise	All	N/A	Fully Substantiated	**Correct**	**Not Redundant**
Limited	**Approximate**	**Some**		Partially Substantiated	Incorrect	Redundant
Wrong	Ambiguous	None		**Unsubstantiated**		
Not Relevant	Cryptic					

Was the Error Framework Correct?

When the results discussed in the preceding paragraphs were analyzed, errors (caused and triggered) were the focus of considerable scrutiny. At each node, error distributions were examined in light of trigger variable settings, degraded ISDs, and because of concurrent testing, the sensitivity test questions. It was this examination that provided the information to verify the error framework. That is, no specific tests were run for this purpose, but all test results were available and contributed to the verification. The analysis of these errors led to several modifications of the error framework. These modifications took the form of software rules that are applied at a subliminal level for the user.

Error Prevention Rules

The error framework defines errors as dependent variables, occasioned by trigger variables. Thus, when a given trigger variable is present, the concomitant errors will also be present. However, as this unfolded in the results, possible exceptions to this relationship were identified. For example, if the Functional Model specified that software applications were not available at a given node, then the addition of software in a simulation would cause the concomitant errors. However, it is more likely that if software tools become available that certain human behaviors would no longer be necessary. Therefore, software applications ought to reduce occasioned errors, not increase them.

This exception led us to establish prevention rules, whereby the presence of certain trigger variables would lead to the prevention of certain errors (i.e., these errors would not be occasioned). It should be noted, however, that must errors (error equivalents of the IV; see Burnstein, 1994, for a detailed discussion) are not prevented by these rules. The prevention rules

associated with software applications are applied globally, as the impact of software applications was felt to be uniform across functions. The errors that are prevented are those most likely to be eliminated by software applications: step-by-step procedures. Table D-4 identifies those errors prevented by software applications.

Table D-4

General Procedural Errors Prevented by Software Applications

Errors prevented by software applications	
GPC1.	Perform the steps incorrectly
GPC2.	Repeat a step when it is not required to do so
GPC3.	Perform an unnecessary step
GPC4.	Perform the steps in the wrong order
GPC5.	Perform a step before there is enough information for doing so
GPC6.	Perform a step too late
GPC7.	Perform a step that is similar or unrelated to the required one
GP01.	Omit a required step
GP02.	Stop the procedure before completing all the steps

A second set of prevention rules was identified for simulations in which the operator variables define highly trained and experienced analysts. In this case, it is more likely that certain errors will not occur, regardless of the trigger variable present, because of the more expert nature of the analysts. The intent is to simulate "expert" behavior in a way that is different from the simulation of less experienced analysts.

The instantiation of these prevention rules is not global but node specific. That is, the effect of highly trained and experienced analysts is not uniform across functions, and functional requirements had to be considered before it could be determined which errors would be prevented. Thus, depending on the node in question, general procedural errors may be prevented or certain hypothesis testing errors may be prevented. Table D-5 identifies the errors prevented by the presence of highly trained and experienced analysts and the nodes to which they apply.

associated with software applications are applied globally, as the impact of software applications was felt to be uniform across functions. The errors that are prevented are those most likely to be eliminated by software applications: step-by-step procedures. Table D-4 identifies those errors prevented by software applications.

Table D-4

General Procedural Errors Prevented by Software Applications

Errors prevented by software applications	
GPC1.	Perform the steps incorrectly
GPC2.	Repeat a step when it is not required to do so
GPC3.	Perform an unnecessary step
GPC4.	Perform the steps in the wrong order
GPC5.	Perform a step before there is enough information for doing so
GPC6.	Perform a step too late
GPC7.	Perform a step that is similar or unrelated to the required one
GP01.	Omit a required step
GP02.	Stop the procedure before completing all the steps

A second set of prevention rules was identified for simulations in which the operator variables define highly trained and experienced analysts. In this case, it is more likely that certain errors will not occur, regardless of the trigger variable present, because of the more expert nature of the analysts. The intent is to simulate "expert" behavior in a way that is different from the simulation of less experienced analysts.

The instantiation of these prevention rules is not global but node specific. That is, the effect of highly trained and experienced analysts is not uniform across functions, and functional requirements had to be considered before it could be determined which errors would be prevented. Thus, depending on the node in question, general procedural errors may be prevented or certain hypothesis testing errors may be prevented. Table D-5 identifies the errors prevented by the presence of highly trained and experienced analysts and the nodes to which they apply.

Table D-5

Errors Prevented by Experience and Training at the Applicable Nodes

Error	Node	Function
GPRC1	1.1.3.1	Determine enemy information requirements
	1.1.3.2	Determine information gaps
	1.2.1	Determine weather impacts on friendly and enemy COAs
	1.2.2.1	Determine terrain impacts on friendly and enemy COAs
	1.2.2.2	Determine weather impacts on terrain
	1.2.3	Determine most probable courses of enemy action
	2.1.1.2	Produce validated requirements
	2.1.1.3	Consolidate requirements
	2.1.1.4	Prioritize requirements
	2.1.2.2	Identify indicators which will satisfy the information requirements
	2.2.1	Determine resource capability and availability
	4.2.2	Identify potential targets
	4.2.3.2	Determine significance of relationships
	4.2.3.3	Produce picture of the battlefield
	4.2.3.7	Determine uncertainties surrounding the COA
GPRC2	1.1.3.1	Determine enemy information requirements
	1.1.3.2	Determine information gaps
	1.2.2.1	Determine terrain impacts on friendly and enemy COAs
	1.2.2.2	Determine weather impacts on terrain
	2.1.1.2	Produce validated requirements
	2.1.1.4	Prioritize requirements
	2.1.2.1	Identify information required for each collection task
	2.1.2.2	Identify indicators which will satisfy the information requirements
	2.1.2.3	Determine enemy nodes, activities, and events that will provide indicators
	4.1.2	Determine if perishable information represents target of opportunity or planned target
	4.2.3.1	Make comparisons between the new information items to determine relationships
	4.2.3.2	Determine significance of relationships
	4.2.3.3	Produce picture of the battlefield
	4.2.3.8	Formulate or disseminate requests for information to obtain clarifying information
APRC1	2.1.1.2	Produce validated requirements
	2.3.1	Perform administration to produce logged specific order and request (SOR)
	4.2.3.8	Formulate or disseminate requests for information to obtain clarifying information
CPC3	1.2.1	Determine weather impacts on friendly and enemy COAs
	1.2.2.1	Determine terrain impacts on friendly and enemy COAs
	2.1.1.2	Produce validated requirements
	2.3.2	Determine current asset capability and availability
	4.2.2	Identify potential targets
	4.2.3.1	Make comparisons between the new information items to determine relationships
HPPRC1	1.2.2.1	Determine terrain impacts on friendly and enemy COAs
	4.2.2	Identify potential targets
HPPRC6	1.1.3.1	Determine enemy information requirements
	1.2.2.1	Determine terrain impacts on friendly and enemy COAs
HPRC1	1.2.2.1	Determine terrain impacts on friendly and enemy COAs
	4.2.2	Identify potential targets

Summary

Through verification testing, it was determined that the computer model functioned as designed, and after some changes, it functions error free. Changes in the computer model resulting from verification testing included some data table changes. Changes in the Functional Model included a reduction in the specified error pools, so that probable or likely errors form those pools, as opposed to possible errors. Changes in the error framework include the specification of error equivalents for degraded information state dimensions, where once there were variable error specifications. Software rules were also developed to take into account certain relationships among trigger variables that had been considered orthogonal.

SENSITIVITY TESTING

Two areas provide the stimulus for sensitivity testing: the information variable and algorithms. In testing the IV, the purpose was to determine if different levels of IV produced different results, that is, resulted in *different* levels of ISDs. The purpose of algorithm testing was to determine if the algorithms produced the *appropriate* changes in the information state dimensions. Sensitivity test questions are summarized in Table D-6. While the tests that addressed these were run concurrently, the inspection of results was unique to each.

Table D-6

Sensitivity Test Questions

Test area	Test questions
Information variable	Do different levels of the IV produce different results?
Algorithms	What are the best algorithm combinations? Do the error impacts on the ISDs demonstrate the appropriate level of sensitivity?

Information Variable Testing

The IV represents the sample of battlefield information that MI acts upon. It is a measure of the degree to which the battlefield information contains sufficient content to satisfy the information requirements of the intelligence producer and intelligence user. The IV is

represented by scalar values of "completeness," whose definitions are provided in Table D-7. Only nodes that receive battlefield information from external sources are impacted by the IV, but that impact is applied consistently across those nodes. Further, errors that result from degraded levels of the IV are must errors (error equivalents that are *triggered* systematically and cannot be corrected during the simulation).

Table D-7

Information Variable Levels

Level	Definition
1	Contains sufficient content to permit intelligence production to meet the user's information requirement.
2	Contains sufficient content to permit intelligence production to substantially meet the most important user information requirements.
3	Contains sufficient content to permit intelligence production to meet some of the user's information requirements.
4	Contains sufficient content to permit intelligence production to begin to address some of the user's information requirements.
5	The content is insufficient to permit intelligence production to meet any of the user's information requirements.

The test method for IV testing was to systematically vary the levels of the IV, changing no other trigger variables, and to inspect results at each node. Thus, these tests were conducted in a way similar to an experimental procedure, with IV level equivalent to the independent variable and the degraded ISDs, the dependent variable. The test criterion was that each level of the IV should produce different ISDs and the ISDs should be ordered according to the IV level (i.e., less complete IVs should produce more degraded ISDs). There were 13 nodes to which the IV applied, and each was tested independently. While there were five levels of the IV, Levels 1 and 2 were not tested as neither trigger errors. Figure D-4 provides the generalized IV test case. Given the concurrent nature of sensitivity testing, IV results were available for both of the two impact algorithms: predominant case and worst case. Since IV error equivalents must be triggered, there is no effect of the error algorithm.

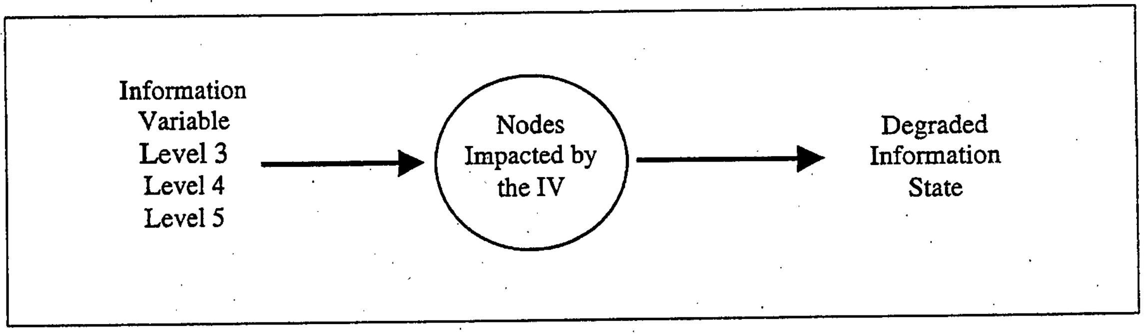

Figure D-4. Generalized test case for information variable testing.

Information Variable Test Results

The different levels of the IV produced different results. However, Level 5 did not produce results supported by the error framework or by face validity; the lack of data did not produce extremely degraded ISDs, as would be expected. This result, coupled with the fact that this level represents an unrealistic situation, caused Level 5 to be eliminated from the IV scale. Given that no differences in the errors were triggered by Levels 1 and 2 (i.e., no errors are triggered in either case), Level 1 was also eliminated. Further analysis indicated that the one error triggered by Level 2 was not in the error pools defined in the Functional Model. While this level is currently ineffective, it was not eliminated, should future work identify a function to which that error applies. Thus, the result of IV testing was that there are two distinct levels of the IV (Levels 2 and 4) that have met the face validity and error framework criteria and will be retained. A third level will be retained (Level 3), but it is untested, as the error equivalent does not occur in the Functional Model.

A second result of IV testing was that the number of IV-applicable nodes was reduced from 13 to 10. Analysis revealed that not all battlefield analysis tasks assumed to be involved were impacted by the information variable. The nodes from which the IV impact was eliminated do not receive collected information directly from external sources but receive evaluated or processed information from previous nodes.

ALGORITHM TESTING

What is the Best Algorithm Combination?

The first algorithm test question addresses the best algorithm combination. The Conceptual Model defines two algorithms that are applied at each node. The error algorithm,

applied to the error distributions, determines errors triggered, based on that distribution. The second algorithm is the impact algorithm, which is applied to the degraded ISDs resulting from error impacts; it determines the final level of the ISD (at that node). Table D-8 provides the combinations that are available from these algorithms. In seeking the best algorithm combination, we will select the optimum setting for each of these algorithms. An optimum error algorithm setting would be one that allows an appropriate change in the ISDs, in relation to the trigger variables. An optimum impact algorithm setting would be one that employs all the information available in the trigger-variable distribution.

Table D-8

Error and Impact Algorithm Combinations

Error algorithm	Impact algorithm
Or	Predominant case
And	Predominant case
75	Predominant case
Or	Worst case
And	Worst case
75	Worst case

A comprehensive approach was used to determine if there was an optimum algorithm combination for each and all nodes. That is, all 34 nodes were tested (individually) with each of the six algorithm combinations. The operator variable, training, was systematically manipulated, as described in verification testing. While this approach yielded results concerning the impact algorithm, the results surrounding the error algorithm were inconclusive. Algorithm testing continued with an approach in which more trigger variable changes and multiple nodes were run, encompassing subsequent errors and providing more information for testing the error algorithm.

This second test approach employed three scenarios constructed by an MI SME and run on the same node clusters. These scenarios helped to increase the face validity of the test environment. That is, a user is more likely to simulate conditions across a broad range of variables than across a single trigger variable. The first scenario simulated a "normal" organization such as a European Division's intelligence processing capabilities with equipment levels fairly high, personnel strengths at about 90%, and good leadership. The second scenario simulated an "above normal" organization with mostly senior personnel and all the automation

and equipment available today. The third scenario simulated a "below normal" organization with poor or non-existent job aids, no automation, poor leadership, and few senior analysts. The results were then analyzed to determine their representativeness of the scenarios. That is, the "best" results should be obtained for the above-normal scenario, the worst results for the below-normal scenario, and intermediary results for the normal scenario.

Algorithm Test Results: The Best Error Algorithm

While the test question addresses the algorithm combination, results were inspected and the determination made individually. Thus, results are reported independently herein. Analysis of error algorithms indicated that application of "or" led to a relatively large number of triggered errors that did not reflect the setup variables. When "and" was applied, the number of triggered errors decreased slightly and was approaching a true reflection of the setup variables. The 75-error algorithm was originally selected as the optimum error algorithm because the number of triggered errors best reflected the setup variables. However, selection of the 75-error algorithm essentially ignores the presence of relevant data. Since the trigger variable distributions contain confidence factors of 50 and 75, application of the 75-error algorithm resulted in consideration of only half of the available data. Therefore, since "and" better reflected the setup variables and uses all distribution data, "and" was selected as the best error algorithm.

Algorithm Test Results: The Better Impact Algorithm

Results indicated that the worst case impact algorithm resulted in change that was too extreme. It did not produce the expected change in the ISD, as reflected by the setup variables and expected by the criteria of face validity. For example, specificity went from precise (highest) to cryptic (lowest) when the expected impact was less extreme (e.g., approximate). The predominant case algorithm resulted in more appropriate change that better reflected the setup variables. This pattern was repeated throughout testing, and it was concluded that the predominant case was the better impact algorithm for all nodes.

The best algorithm combination, then, consisted of predominant case and "and." This combination produces results that are most reflective of the simulated conditions and uses the information available in the error distributions.

Do the Error Impacts on the ISDs Demonstrate the Appropriate Level of Sensitivity?

The second algorithm test question addressed the impact of errors on the ISDs and asks if they are appropriate. That is, if the errors occurred in the "real world," would one expect the information to be degraded to the extent observed? For example, if only one error is triggered

(e.g., more data were collected than were needed) and completeness is observed to degrade from "all" (highest scalar value) to "none" (lowest scalar value), the impact of that one error is probably too great. Likewise, if many errors occur, or a small number of significant errors and little change in the information are observed, the impact is probably too small.

The purpose of this testing was to inspect the degree of change observed in the information state dimensions (as a result of error impacts). The question of change, relative to ISD scalar values, had been of interest because of the nature of those scales. The trigger variable scales are nominal scales that differ greatly across the ISDs. Table D-9 presents the ISDs and their levels. As seen in this table, two scales (accuracy and redundancy) are comprised of only two levels that yield dichotomous results, and except for completeness, the scales are not balanced. The remaining scales are negatively loaded. Because of these scalar attributes, it is possible that considerable change could result from relatively few or minor errors, particularly for the dichotomous dimensions.

Table D-9

Information State Dimensions

Relevance	Specificity	Completeness	Perishability	Validity	Accuracy	Redundancy
Relevant	Precise	All	Lasting	Fully substantiated	Correct	Not redundant
Limited	Approximate	Some	Temporary	Partially substantiated	Incorrect	Redundant
Wrong	Ambiguous	None	Transient	Unsubstantiated		
Not relevant	Cryptic		Elapsed			

Results were inspected, again, from a face validity standpoint of whether such changes could be expected in the real world, given the simulated conditions. The test approach embodied by the three scenarios was used and allowed analysts to compare simulation results with "known" results from the performance of MI organizations similar to those simulated. Results from other, independent node tests were also available for inspection.

Results of Testing Impacts on the ISD

Results of this testing repeatedly displayed extreme degradation for relevance, specificity, and validity, regardless of the number or type of errors triggered. Accuracy and redundancy, because they are dichotomous dimensions, also posed a problem: any change was extreme. Completeness changed only from "all" to "some," never degrading to the full extent ("none"). Finally, perishability degraded only to a small degree. These results confirmed initial suspicions that the ISD scales would require modification.

Except for accuracy and redundancy, these modifications took the form of expanding the scales to five or six levels. These expanded scales, shown in Table D-10, enable more appropriate, moderate degradation of the ISDs. As a result of this testing and continued analysis, both dichotomous dimensions (accuracy and redundancy) were eliminated. While this attribute contributed to their elimination, it was not the sole determinant. After continued analysis, it was determined that redundancy was an irrelevant dimension. That is, redundant information is not an important consideration for MI production. It is frequently present (e.g., multiple battlefield reports of the same event), but it need not be in order for the information to be successfully processed. Accuracy, as it was defined (correctness of output relative to input) is a somewhat indistinct dimension for a system whose purpose is to transform information and thus perhaps obscure "correctness." Analysis revealed that this dimension too was irrelevant to MI production.

Table D-10

Expanded Scalar Values for Information State Dimensions

Relevance	Specificity	Completeness	Perishability	Validity
Fully relevant	Precise	All	Lasting	Fully substantiated
Mostly relevant	Precise, with additional analysis	Most	Temporary, little impact	Mostly substantiated
Limited, adequate	Approximate, useful	Some, sufficient	Temporary, adequate	Partially substantiated, sufficient
Limited, insufficient	Approximate, with major gaps	Marginal	Transient, some utility	Partially substantiated, insufficient
No relevance	Ambiguous	Insufficient	Transient, little utility	Unsubstantiated
Wrong relevance	Cryptic		Elapsed	

Summary

Sensitivity testing led to changes in the Logical Model that resulted in an appropriately sensitive computer model. These changes included the reduction in the levels of the information variable from five to three and the elimination of the IV impact from three functional nodes. Further changes included elimination of two information state dimensions and the expanding of the scales for the remaining five ISDs. Also resulting from sensitivity testing was the selection of the most appropriate algorithm combination: predominant case and "and."

APPENDIX E

THE MILITARY INTELLIGENCE CONCEPTUAL MAP

THE MILITARY INTELLIGENCE CONCEPTUAL MAP

INTRODUCTION

Recent advances in technology have changed the way that MI does business. At the same time, the changing military is operating under austere budgetary conditions, undergoing realignment, and facing significant changes in doctrine. It is critical that the impact of these changes on intelligence production be understood. The most cost-effective manner of assessing these impacts is through modeling and simulation. The IPPM was developed to assess the impact of change on intelligence.

This appendix describes the MI Conceptual Map, a framework developed for the purpose of modeling MI production performance from the perspective of information, the currency of exchange in MI. Earlier appendices described the process to simulate human performance in the intelligence production system. This appendix describes the process by which content-free information is modeled. The first section in this appendix explains the framework employed: conceptual mapping. The second section describes how information is used in the computer simulation model, that is, how information flow among human tasks is instantiated and how it is measured. The final section explains how integrating the Performance Model and the Intelligence Conceptual Map expanded the Logical Model (see Appendix C).

BACKGROUND

Inherent in the intelligence production system is the ability to improve the worth of collected data through the synergy of the analytical process. Disparate bits of data are integrated in a process that achieves an information product whose value may be greater than the simple sum of its parts.

Because the performance model was designed only to address the impact of errors on input information, its logical model contained no capability to address this facet of the domain. Consequently, it was necessary to expand the scope of the original conceptual model to include the simulated flow of information.

INFORMATION FLOW

Just as the performance module required a functional representation of the domain, the expanded concept required an information representation within the domain. Further, this

conceptual requirement implied a need to measure information flow while maintaining the input-process-output paradigm and the content-free environment of the performance module. As indicated earlier, most previous research work in this regard concerned the flow of actual information, and measurements were oriented on efficiency, mainly from a timeliness viewpoint; the decision was made to pursue the task from a content-free effectiveness view.

EXPANDED CONCEPTUAL MODEL

The expansion of the conceptual model was simply the next logical step in the development progression. In its new form, it provided a guide to identify the elements, measures, and sequencing necessary to simulate information flow in the intelligence production system. The addition of simulated information flow allowed us to predicate the impact of information quality on performance and to provide better definition to the impact of performance on information quality; further, this approach allowed an ultimate judgment of the "value" of intelligence in the military domain.

General

The requirement to "objectify" the understanding of information in the domain led us to explore the idea of concept mapping. Concept mapping, alternatively known as cognitive or knowledge mapping, is a method widely used in education and business to represent knowledge as a spatial network of concepts in which interrelationships are specified (Al-Kunifed & Wandersee, 1990; Ausubel, 1968; Bernard & Naidu, 1992; Cossette & Audet, 1992; Donald, 1987; Eden, 1994; Finley & Stewart, 1982; Gillan, Breedin, & Cooke, 1992; Gordon, Schimierer, & Gill, 1993; Lambiotte, Dansereau, Cross, & Reynolds, 1989; Leong, 1992; Novak & Gowin, 1984). Conceptual maps can be used to represent both domain-defined and idiosyncratic knowledge of the domain. Such maps are typically composed of structures, either circles or squares, connected by links that may or may not have arrows indicating direction; these links usually include a verb that identifies the relationship between the structures. As a rule, movement from the lower levels of the concept map through the upper levels means a graduation from detailed, specific information to more conceptually oriented, general understanding.

Conceptual Map in the MI Domain

To begin the development of the MI Conceptual Map, hereafter referred to as the ICM, prior research (Burnstein, Fichtl, Landee-Thompson, & Thompson, 1990) required the users of intelligence to specify their information requirements, and a domain-oriented taxonomy was

employed. Our MI SMEs used the information requirements hierarchy defined in earlier work to begin developing the semantic network that would eventually become the ICM. The conceptual network constructed was intended to be a normative domain representation of the hierarchical understanding (accumulated from data collected to the ultimate domain goal) about the enemy in the future. As information items not included in the table were identified, they were added to achieve the final conceptual map. There are three separate but interrelated vertical chains in the ICM; these extend from databases at the bottom to information about the future at the top. They represent information about and understanding of the distinct categories of enemy, friendly, and physical environment and the relationships among the three. The enemy and friendly hierarchies are mirror images of one another, so the friendly chain was not expanded because this research was enemy oriented. The individual nodes in each chain represent information about and understanding of a specific aspect of those subject areas; the definitions of the individual nodes in the ICM are provided in Appendix F. The links between the nodes represent two different kinds of relationships and are defined as follows:

1. Data provisional relationships. The direct association of two or more nodes in a parent-child relationship in which new information is coalesced from children to parent, and the whole may be greater than the sum of its parts.

2. Transformational relationships. The indirect association of two nodes in a relationship in which no data flow, but understanding in one is a catalyst to understanding in the other.

For example, the ICM shows that information in the enemy capability and intent nodes leads to understanding in the enemy mission node, and understanding of current enemy activities supports understanding of enemy forces.

The ICM was developed as the domain element to fulfill the conceptual requirement for instantiation of information in the simulation of the intelligence production system. However, standing alone, it did not provide any means to represent and measure the flow of information.

Extension of the ICM

In order to represent the flow of information in the ICM, a number of additional concepts needed to be developed. These were required to account for the changes in the outside environment, the collection of data, the measurement of information quality, and the satisfaction of user requirements. These issues were addressed, respectively, by the concepts of operational goal, asset suites (a group of collectors) related to databases, information quality expressed in terms of dimensions,

and intelligence requirements (IR). The addition of these elements in the ICM provided the mechanisms necessary to accumulate and represent information quality without regard to process.

Operational Goal

The operational goal (OG) is the overarching pragmatic frame that inserts change into the ICM. The OG imposes the circumstances of a specific situation on the map; all military aspects of a situation that might differ in a specific scenario are accounted for. Our intent was that the combination of alternatives for the several facets of OG would determine environmental conditions, user needs, IR, and the composition of the asset suite.

Asset Suite

In the intelligence processing system, data are collected by people and technical systems that are normally associated with an Army echelon or level of organization. A generalized set of collection assets by echelon was developed. In typical circumstances, the alternatives of OG would cause the selection of one of these suites. There was a belief that the individual assets in each suite were capable to varying degrees of contributing data in a normed set of attributes that describe all data that might be collected. These attributes are shape, size, quantity, presence, absence, dynamics, parametrics, and human dimension (all are defined in Appendix F). This idea was extended by proposing that the combined attribute contribution, in aggregate over an entire asset suite, varied in relation to the specific circumstances of the scenario and to the method in which assets are employed. This contribution was represented by information dimensions (completeness and specificity) for a set of information attributes. Conceptually, this established a connection between the collection assets and the ICM and provided the initial measurement of information quality.

Databases

Databases are the initial repositories of information in any information processing system. Three primary databases, organic to the ICM, support the three individual hierarchies within the map: enemy, physical environment, and friendly. These databases were decomposed to their most elemental levels, the bottom level of the combined database being a single node that was named the central database. This node represented the lowest level to which data can be decomposed and still include all that can be understood about any given increment of data. It was determined that it represented the information attribute level of data and was divided into

behavior, spatial, temporal, structural, and quantity categories (BSTSQ) (defined in Appendix F). Databases were decomposed in this manner for two purposes. The primary purpose was to achieve a level of decomposition that would allow direct association with the attributes of collection. Secondly, the subdivision provided a representation that accounts for knowledge that exists at the initiation of an operation; this portion of the database was labeled historical. The latter was required because there can never be a situation when the databases are empty; at a minimum, data must exist in the friendly and physical environmental hierarchies.

Quality

Understanding at any given node was determined to be the quality of information residing in that node at any point in time. The information state dimensions described in Appendix C were revisited, and it was decided that the flow of information quality in the ICM was best represented by the dimensions (completeness and specificity). Because of the nodal relationships in the ICM just described, completeness and specificity could be defined for each of the five information attributes (BSTSQ) at every node in terms of the ordinal measurements—the elements that are also defined in Appendix F.

Information Required

Similarly, IR is the representation of context-specific user needs expressed in the same terms of quality measurement. These represented the user's operational requirements that would logically reside in each of the nodes of the ICM above the database level. The operative thesis that was developed for any given situation was that the user of intelligence would have distinct requirements for information at specified nodes in the map; it was concluded, as well, that these requirements would be dictated by and changed with the circumstances of the situation and environment. Once quality requirements were established in any of these nodes, resultant requirements could be represented at all other nodes because of the relationships among the nodes, which were discussed earlier.

THE ICM UTILIZATION FRAMEWORK

In order to actually use the ICM as part of the computer simulation model, there was a requirement to value the contribution of information developed by the intelligence production system; the concepts described earlier included information value but did not provide the mechanisms that would allow either information or its surrogate to move within the map nor was

there a system that would produce a valuation of the contribution resulting from that movement. The next requirement, therefore, was to develop those mechanisms to instantiate information flow and its valuation in the ICM and then in the computer model.

MEASURING VALUE

Constraints

Being constrained by the necessity to use prior development efforts when possible, the next requirement was to develop information value mechanisms in a manner that would be generally compatible with the performance module (PM). This meant that the ICM had to model information in such a way that it was compatible with the content-free environment that already existed in the PM. As indicated earlier, it was determined that an acceptable vehicle to represent information in such an environment was information quality that was clearly subject to being improved or degraded while being devoid of content.

Information Quality

Having already established that data would be described in terms of the attributes of behavior, spatial, time, structure, and quantity, the next development related to the terms in which the quality of these attributes (and indirectly that of information) would be expressed. As indicated before, it was previously decided to represent data collected by the completeness and specificity dimensions. The performance module dimensions for an information state were investigated; these were relevance, specificity, completeness, perishability, and validity. After analysis, it was concluded that only completeness and specificity were needed to adequately represent the quality of information within the ICM. Relevance and validity were eliminated because the internal relationships within the map generate an accumulation of understanding that makes them unnecessary; perishability was eliminated because of the absence of elapsed time in the ICM. It was determined that the completeness and specificity dimensions were appropriate measures of quality in the conceptual map because they adequately represented the worth of information to the user of intelligence.

The ordinals used to measure each of the completeness and specificity dimensions are defined in Appendix F. The PM scale for specificity was altered for symmetry reasons by combining levels for ambiguous and cryptic into a single level. Therefore, for any elemental data bit or developed information about a given subject, there exists a complete set of ordinals in each

of the dimensions (completeness and specificity) for all of the attributes, BSTSQ, which describes the quality of that data or information.

Populating the Database

The intelligence processing system populates its databases by means of a collection system composed of people and machines. These collect "data" in various forms and add them as input to databases that already exist, as with the historical portion of the database. A structured, normed collection system had already been developed that was represented as an asset suite and described what could be collected in terms of the attributes of collection: shape, size, quantity, presence, absence, dynamics, parametrics, and human dimension (see Appendix F). It remained to establish a connection between these attributes and the attributes of information (BSTSQ). It seemed logical to assert that any given data collector could be valued with regard to its potential to provide input to the attributes of collection. This logic was extended to an entire group of collectors, called an asset suite, and represented this capability in terms of ordinals for completeness and specificity in the same way data quality was described. It was further believed that the attributes of collection were directly related to the attributes of information and could be converted, one to the other, when both were measured in similar terms. In this manner, a content-free, quality relationship expressed in terms of completeness and specificity was established. Because the lowest level in the ICM, central database, and the upper level of the collection function were both represented in the quality dimensions (completeness and specificity), a quality measurement relationship between the collection function and the bottom of the ICM was also established.

Representing Nodal Quality in the ICM

Once quality was established for the initial node in the ICM, the development of a straight-forward, combinatorial approach for establishing like quality representations in the rest of the map was the logical next step. Based on the upward progression of relationships among the nodes, a similar representation of quality for every node in the ICM was established. In effect, these measurements represent the quality of data collected and information developed within a perfect processing system because the processing errors occasioned by the PM have not yet been applied.

Value in the ICM

At this point in the development process, quality in the ICM was represented from two distinctly different points of view. The first, as discussed earlier, was the quality of information

needed by the user of intelligence to satisfy a particular operational requirement; the second was the quality of information collected by the asset suite and combined with other information in the database. With these two quality representations residing in each of its nodes, the ICM reflected a measurement of value; in effect, the quality of information collected and improved through analysis was compared to the quality of information required to accomplish generalized functions. At this point, the degrading of quality, although possible, had not been applied because the model logic still was assumed to be a perfect or errorless process. This comparison provided a valuation of the intelligence product by demonstrating whether the user's informational needs at specific nodes were met. At least in part, this comparison provided a systemic "value added" answer.

RELATING THE PM AND THE ICM

Intrinsic in conceptualization of any information processing system is the necessity for interaction of impacts between performance by the human operators of the system and the quality of information upon which they act. In practical terms, that meant there needed to be a consideration of how the quality of data input would affect performance as represented by the errors of the PM and how those errors would affect information quality in the ICM. This portion of the development began with the dual assumptions that good quality would lessen the likelihood of error and that absence of error would not improve low quality.

ICM "is" Relationship With the PM

The ICM is a conceptual map of a continuous, iterative process that has been represented at a point in time. In considering the integration of performance and information quality, it was concluded that the ICM existed in its entirety, relative to the PM, at all times. Further, it was believed that a support relationship between ICM and PM nodes existed (i.e., that the information residing in the nodes at a given level of the ICM was needed to support the performance within specific nodes of the PM). Of course, in simulating the intelligence process, both performance and information were instantiated as a generalized representation of errors and information quality. With this generalization, the individual nodes of the PM and ICM were associated in order to identify the mapped locations of opposing impacts. The individual processing steps of the PM are supported by the individual informational nodes of the ICM.

THE LOGICAL MODEL EXPANDED

The final step in developing a simulation of the intelligence process was to expand the logical model by integrating the PM and the ICM so that the interaction of the two functional components would determine the quality of information required and produced within a user-specified context. In effect, this would provide a more complete simulation of the impact of change on the information processing system. This effort was bound by the same assumptions and constraints discussed in the introductory portion of earlier sections; therefore, the expansion of the logical model was intended to more definitively describe components of the intelligence processing system and prescribe rules that would help determine information quality. The model remained a content-free, generalized representation of any Army situation across the spectrum of conflict.

THE APPLICATION OF OPERATIONAL GOAL

Following this intent, OG was compared to the operation parameters of the PM. Since the individual OPs were similar in concept to the individual facets of the ICM's operational goal, those that were appropriate to setting ICM situational conditions were used without alteration. In this way, the same circumstantial change was imposed in both the PM and ICM, thereby achieving a considerable degree of sameness in the two modules. The OPs were adopted essentially unchanged with regard to the operation of the performance module, but only those OPs necessary to operation of the information module were used on that side. Additionally, as requirements for more ICM OPs were identified, they were developed and added to the logical model (new OPs are defined in Appendix F).

To instantiate the OPs in the ICM, a series of rule sets was developed that would collectively apply their individual alternatives to the information module. Their application produced the capability to vary the content and output of the information module because they drove changes in user requirements, initial values for the historical database, and the constitution of the asset suite.

APPLICATION OF OPPOSING IMPACTS

As indicated earlier, it was known that both the performance and informational sides of an information processing system had opposing impacts on one another. This led to the requirement to develop representations that would apply the effect of information quality to the occurrence of

performance errors and would apply the impact of errors committed to the quality of information produced.

Analysis of the PM revealed that modification of its concept for the information variable was an appropriate means to improve the representation of information quality impacts on the incidence of performance error. The PM has no rules or guidance to determine the IV level; as a result, IV was arbitrarily set without reference to what the information value might be in a specific situation. The decision was made to use the information quality values for the dimensions (completeness and specificity), residing in specific nodes of the ICM to set IV for related PM nodes; this was a much more effective way to represent the influence of information quality on performance. The degree of effectiveness had been increased both by specifying the related nodal locations and by representing the developed information value rather than using an arbitrary setting.

The process for applying error impact to information quality was somewhat more complicated. It was decided that error impact should be applied at specific information nodes, based on the occurrence of errors as logical entities within functional groupings of nodes in the PM. Once an error grouping was accumulated, the error impacts would be determined by rules that assigned quality degradation to specific nodes, based on the nodal associations. As with quality impact on performance, this was a better representation of the process because the impact of performance on quality is specified as to location within the ICM and it is actually applied to a value that has been developed within an operational context.

SENSITIVITY ALGORITHM

Sensitivity in the PM was related to the likelihood of errors being committed. For the information module, it was decided to make sensitivity dependent on the impact of errors on information quality in relation to the quality of databases. It was recognized that full development of this concept could only be accomplished during sensitivity testing.

VALUE IN THE ICM

Earlier sections described information value in terms of a perfect or errorless process. With the addition to our logical model of the concepts for the cross application of the effects of information quality and errors, the valuation of processed information with regard to the user's requirements had been achieved. The logical model had been developed to the point where it

instantiated the various facets of the intelligence processing system so that good and bad performance of good and bad information could be represented. More importantly, a logical model had been conceived in which quality could be improved as well as degraded.

The instantiation of OG by setting OPs established the scenario. Algorithm settings established model sensitivity. Scenario parameters established the initial content of databases, what collectors are used, the information requirements of users, and the patterns of performance error that might occur. These various elements established all of the representations of information needed to simulate the value of information collected, processed, and used; in other words, the entire intelligence processing system was represented.

In this context, the value of intelligence in the logical model was revealed by and should be measured at those nodes that are directly related to the PM nodes that provide output to a user. In effect, it was proposed that value be represented by a comparison of information collected and processed to the user's requirements in a specified situation.

APPENDIX F

DICTIONARY OF TERMS AND DEFINITIONS

DICTIONARY OF TERMS AND DEFINITIONS

INFORMATION ENVIRONMENT VARIABLE

This is a measure defined in terms of completeness of the degree to which a sample of information from the battlefield contains the information elements necessary to satisfy the intelligence producer (internal) and the intelligence consumer (external). This independent variable is used to represent systemic errors in the intelligence production system, regardless of processing performance, when inadequate information is used.

1 and 2. Contain sufficient content to permit intelligence production to substantially meet the most important information requirement.

3. Contains sufficient content to permit intelligence production to meet some of the information requirements.

4 and 5. Contain sufficient content to permit intelligence production to begin to address some of the information requirements.

Note. This variable is not set by the user. It is one of the output data points in the detailed reports, which may be responsible for degradation in the quality of information because of data sources. It is reported in the detailed error report as information variable must error (IVME).

SENSITIVITY·ALGORITHMS

Sensitivity algorithms provide the capability to affect modeling parameters that specify the degree with which production performance and information quality respond to influences or control variables in the system being modeled. The error occurrence sensitivity algorithm applies to production performance, and the effects of errors on intelligence products apply to information quality. The levels of sensitivity are low, medium, and high for the first, and major, medium, and minor for the latter.

Error Occurrence

One component of the IPM models production performance in terms of functions or processes performed by analysts, control variables (work environment and task) that influence the performance of these functions, and errors with a potential to occur because of the influence

of these control variables on human performance. Error potential plus sensitivity to the potential for an error to occur determines whether an error actually occurs or is "triggered."

Each control variable set for a particular function contributes uniquely to the potential for one or more errors to occur in that function's processes. It may contribute to one of five possible degrees of potential: 0, 25, 50, 75, and 100. For each degree of potential, the sum of contributions is accumulated for each function and is compared to a pre-set threshold for that function. The sensitivity algorithm looks at which and how many of the degrees for error potential have exceeded the threshold. Depending on the level of sensitivity set for the model, errors may or may not be triggered. For example, "low" sensitivity means more than one degree of potential must be exceeded.

When modeling a particular scenario, this algorithm is used to describe some aspect of the unit's performance expectation. For example, when analysts are mostly experienced and well trained, they are less likely to actually commit an error even when the potential is high.

Effects of Errors on Intelligence Products

Another component of the IPM models information quality in terms of its measure of completeness and specificity (see Information Quality Measures and Dimensions). This is done in two phases. First, information quality is determined according to what was contributed by its data source, independent of the impact of performance errors. Next, the model logic recognizes that the quality of data and information used by analysts to produce intelligence influences the potential for error (see Information Environment Variable Definitions) and further recognizes that errors in performance may further impact the quality of information and intelligence.

The effect of errors on information, that is, the intelligence products, is determined according to the sensitivity of those data to errors. Errors in groups of functions, such as all functions related to collection management processes, are accumulated, and the sensitivity algorithm applies the impact to the quality measure for the associated information. The degree of the impact is determined by the setting of this algorithm. Errors that occurred in the collection management functions affect information quality in the database nodes, for example.

When modeling a particular scenario, this algorithm is used to describe an expectation that information and intelligence products may or may not respond to performance errors. For example, when information in the historical databases is known to be sketchy and of low quality,

errors in the production activities, which transform these data into richer information, would have a greater impact on intelligence products than if these data were very good.

Note. This variable is not currently set by the user. These variables are automatically set at low and minor for "error occurrence" and "effects of errors on intelligence products," respectively. There will be variations of these settings reflected on some older test cases, and the model developers are able to establish test cases using other variations in these settings.

SCENARIO ENVIRONMENT VARIABLES

These provide the means for specifying the operational environment in which the simulation will run. Scenario environment parameters were developed by decomposing aspects of the operational environment thought to have an effect on the modeled MI behaviors. Scenario environment settings are a method for inserting a general scenario into the model process. The model has 17 different parameters within these categories, which can be used. These parameters also determine default settings for the MI task personal and performance variables using a precedence system. There are six relevant aspects of the operational scenario: operational, mission, soldier, battlefield, task, and collection environments.

Operational and Mission Environment

These parameters describe salient operational and mission-related aspects of the operational environment.

Level of War

This operational parameter defines the entire spectrum of military operations for both warfare and operations other than war (OOTW). As a rule, the higher the level of war, the higher the echelon; this does not preclude a user from setting an echelon that differs from the level of war.

Tactical	Execution of operations to win battles and engagements that are near term and have relatively immediate consequences.
Strategic	National, alliance, or coalition objectives.
Operational	Planning and executing campaigns that further strategic objectives.

Battlefield Operations

This operational parameter describes the predominant characteristic of the battlefield, based on the operational continuum.

Other Than War	For any operation other than conflict, including humanitarian missions, counter-narcotics, peacekeeping operations, and so forth.
Nonlinear Battle	For low intensity conflict (LIC), middle intensity conflict (MIC), or high intensity conflict (HIC) that follows the nonlinear pattern of a lack of well-defined close and rear battlefields, high mobility, high tempo, and a deep attack.
Linear Battle	For conflict (LIC, MIC, HIC) that follows the more traditional pattern of a well-defined battlefield in terms of close, deep, and rear; maintains a clear-cut delineation between offensive and defensive operations.

Force Composition

This operational parameter allows the modeler to choose one of two force compositions.

Joint or Combined	Joint operations are conducted by two or more or the Armed Forces of the United States. Combined operations are conducted by forces of two or more allied nations acting together to accomplish a single mission.
U.S. Single Service	Operations are conducted by one branch of the United States Armed Forces.

Echelon

This mission parameter defines the organization level of focus.

Theater	Division
Army	Brigade
Corps	Battalion

Support Relationships

This mission parameter characterizes the support relationship between the MI's organization that performs the intelligence production process and its controlling headquarters. Options are based on the familiarity of operating within the headquarters.

Habitual	A relationship that exists most of the time when the intelligence staff habitually supports the controlling headquarters organization. Standing operating procedures (SOPs) are understood and the intelligence staff habitually uses and understands these SOPs, whether written or unwritten, in satisfying command intelligence requirements.
Some Past Relationship	Although the relationship is not continuously habitual, the intelligence staff has worked with the controlling headquarters sufficiently to understand most SOPs, and the working environment is familiar.
New Non-habitual	This is the first time the intelligence staff has worked with the controlling headquarters. SOPs are unfamiliar and the procedure must be learned.

Type of Mission

This mission parameter describes the predominant character of the mission.

Move	Any operation in which movement dominates; both movement to contact and retrograde are examples.
Defend	Any operation in which defense dominates.
Attack	Any operation in which offense dominates.
Conduct	Any operation in OOTW ("execute" in an alternative).

Soldier Environment

These parameters describe both physical mission-oriented protective posture (MOPP) level and non-physical (stress and morale) transient operator characteristics.

MOPP Level

This parameter allows the modeler to set a MOPP level. Options are categorized according to standard military MOPP levels.

MOPP Levels 0 and 1	Mask, gloves, and boots are carried; clothing is worn or available.
MOPP Level 2	Clothing and boots are worn; mask and gloves are carried.

MOPP Levels 3 and 4	Clothing, boots, and mask are worn; gloves are carried or worn.

Stress

This parameter allows the characterization of the stress level of the unit or section. Selections are based on the modeler's perception of stress levels.

High	Describes the condition of excessive stress that would be expected to have a profound and immediate effect on performance.
Moderate	Describes the condition of slightly more stress than is customary. Prolonged exposure at this level of stress may have an impact on performance, but the impact may not be profound or immediate.
Minimal/Normal	Describes either the condition of lack of stress or of routine stress.

Morale

This parameter allows the characterization of the morale of the section or unit. It is not tied to the level of stress, recognizing that some levels of stress have beneficial impacts.

High	Describes the condition wherein the majority experiences a high level of esprit, unit pride, cohesion, and camaraderie.
Average	Describes the condition wherein the majority routinely experiences individual pride but will exhibit teamwork when required.
Low	Describes the condition wherein the majority feels isolated as individuals; there is no esprit and little to no teamwork.

Battlefield Environment

These parameters describe tangible aspects of the battlefield.

Battlefield Conditions

This parameter describes the physical environment in which the intelligence production process will operate. These options come from those described in Field Manual (FM) 100-5, Operations. The modeler chooses the one option that generally captures the

operating environment. If more than one option fits the description, then the modeler needs to select the option that the organization has the most potential to operate in or make more than one run of the simulation.

Arctic/Winter	Temperatures remain below zero for extended periods of time. Bodies of water and ground are usually frozen.
Desert/Arid	Weather conditions are excessively dry and can change rapidly. Temperatures range from 30° to 130° Fahrenheit in a 24-hour period.
Temperate	Weather conditions are moderate.
Rain Forest/Jungle	There is thick vegetation, constant high temperature, heavy rainfall, and humidity.
Mountainous	A land mass that makes maneuvering difficult. The weather can vary.
Urban	The battlefield is in a city setting.

Physical Environment

This parameter describes the condition in which the soldier works.

Fixed Facilities	These are buildings or semi-permanent structures that afford protection from the elements and from enemy fires—considered to be relatively safe
Tracked Vehicles	These include MIL vans as well as command and staff type tracked vehicles. These provide a fairly safe and stable working environment with fairly adequate workspace and some temperature control.
Tents	Although tents provide some protection from the elements, the protection is minimal.
No Shelter	No protection from the elements other than the clothing the participant wears.

Task Environment

These parameters describe the resources available to the unit for performing their tasks.

Reference Materials Available

This parameter describes the sources of information, other than the sensor data and intelligence, available to the scenario. It includes those information sources that aid the soldier in performing his or her job, such as field manuals, SOPs, message templates, and so forth.

All/Most	More than 85% of required references are available and used to perform a function.
Some	Between 50% and 85% of required references are available.
Little/None	Fewer than 50% of the required references are available.

Intelligence Systems Maturity

This parameter describes the maturity of the intelligence processing systems available to the scenario. The categories listed below are designed to capture the emerging MI "revolution" in processing capabilities. Examples of established versus developmental systems include TRAILBLAZER and the ground-based common sensor. The categories of documented and undocumented are meant to convey the extent of documentation (operator's manuals, maintenance manuals, etc.) provided with a system or system modification. Examples of documented versus undocumented systems might include Microfix and HAWKEYE-Warrior, respectively.

Established Systems Documented	This variable indicates that the system being used has completed development, and the documentation for the system is current.
Established Systems Undocumented	This variable indicates that the system being used has completed development, but the documentation for the system is not current or complete.
Developmental Systems Documented	This variable specifies that the system being used is still in the development phase, and the documentation is as current as can be expected.
Developmental Systems Undocumented	This variable specifies that the system being used is still in the development phase, and the documentation is not current.

Manpower

This parameter refers to the number of soldiers available to perform the required tasks.

More Than Enough Interfering	Describes the situation that occurs when there are so many soldiers assigned to a task that efficiency degrades.
More Than Enough	Describes the situation of having excess soldiers with no detrimental effect.
Enough	Describes the situation of having a sufficient number of soldiers.
Less Than Enough	Describes the situation of having fewer soldiers than is optimal but without detrimental effects.
Less Than Enough, Interfering	Describes the situation of having insufficient soldiers with a detrimental effect on the ability of the section to accomplish its task.

Mission-Related Training

This parameter is a subjective evaluation of the training status of the organization performing the functions in the intelligence production process. The evaluation is generalized and stated in terms of training status indicators.

Untrained	The organization performing the functions in the intelligence production process has worked and trained together to a proficiency of less than 50% of the required collective tasks.
Partially Trained	The organization has worked and trained together to a proficiency of less than 80% but more than 50% of the required collective tasks.
Trained	The organization has worked and trained together to a proficiency of more than 80% of the required collective tasks.

Collection Environment

These parameters describe the information-providing resources available to the unit.

Availability of Friendly Data

This parameter describes the quantity and quality of friendly data available for the organization's area of operations.

Some	Some of the friendly data are available.
All	All inclusive.

Availability of Background Data

This parameter describes the quantity and quality of background data available for the organization's area of operations.

All	All inclusive.
Most	Most of the background data are available.
Some	Some of the background data are available.
None	None of the background data are available.

Assets

This parameter enables the user to specify from which echelon the basic set of collection resources will be drawn, recognizing that assets available in a mission are not always (or even usually) the same as the operational echelon.

Theater	Division
Army	Brigade
Corps	Battalion

PERFORMANCE NODES

The tasks and functions that represent the intelligence production process are modeled in the IPM as a nodal structure of inputs, processes, and outputs. There are 34 nodes, identified by a functional decomposition of typical intelligence production functions. This nodal structure is depicted in the User's Manual. Each of the 34 functions is defined next. The integer is an identifier used in the input screens of the model. The engineering notation describes the major function identifier and subsequent decomposition of sub-functions, in which 1.0s are battlefield area analysis functions, 2.0s are collection planning functions, 3.0s are collection operations functions[2], and 4.0s are analysis and production functions.

[2]Collection operations functions (3.0) are not modeled as analyst tasks; rather, the results of collection operations are modeled by the assets and asset ratings modules in the IPM.

Node ID	Function ID	Node Description
1	1.1.1	Determine weather information requirements
2	1.1.2	Determine terrain information requirements
3	1.1.3.1	Determine information requirements for battlefield planning
4	1.1.3.2	Determine information gaps
5	1.2.1	Determine weather impacts on FR and enemy COAs
6	1.2.2.1	Determine terrain impacts on FR and enemy COAs
7	1.2.2.2	Determine weather impacts on terrain
8	2.1.1.1	Perform requirements administration to produce intelligence requirements
9	2.1.1.2	Produce validated requirements
10	2.1.1.3	Consolidate requirements to combine like requirements
11	2.1.1.4	Prioritize requirements to produce prioritized list
12	2.1.2.1	Identify information required for each collection task
13	2.1.2.2	Identify indicators that will satisfy information requirements
14	2.1.2.3	Determine enemy nodes, activities, and events that will provide indicators for SIRs
15	2.2.1	Determine resource capability and availability
16	2.2.2	Prepare SORs
17	2.3.1	Perform administration to produce logged SOR
18	2.3.2	Determine current asset capability and availability to produce specific sensor selection
19	2.3.3	Develop asset employment plan
20	2.3.4	Oversee collection mission to produce SOR response
21	2.1.3	Evaluate response to produce separated critical information needs
22	4.1.1	Identify and disseminate force protection information
23	4.1.2	Determine if perishable data represent a valid target
24	4.2.1	Produce new or updated data records for situation and target development
25	4.2.2	Identify potential targets
26	4.2.3.1	Make comparisons between the new information items to determine their relationships
27	4.2.3.2	Evaluate enemy relationships against known relationships to determine significance
28	4.2.3.3	Analyze locational data, current activity, composition, and combat effectiveness of enemy forces to produce battlefield uncertainties and enemy situation
29	4.2.3.4	Identify existing indicators of possible ENCOAs
30	4.2.3.5	Identify possible ENCOA
31	4.2.3.6	Wargame enemy course of action to determine most likely
32	4.2.3.7	Determine uncertainties surrounding the course of action
33	4.2.3.8	Formulate and disseminate requests for information to obtain clarifying or missing information
34	1.2.3	Determine most probable ENCOA

PERSONAL AND PERFORMANCE VARIABLES

These are operator and operational variables that represent changes in the situation being modeled. Operator variables represent aspects of human performance brought to the situation, while operational variables represent aspects of the situation outside the operator and define the conditions during which operators must perform.

Personal Variables

Knowledge Variables

Knowledge is derived through experience and training. It can be estimated by a composite test score based on individual measures, the expected level of training given military occupational status and grade, and experience based on assignments. Since it is difficult to derive possible errors from test scores, training and experience are further defined.

Level of Training:

Entry Level	Represents the formal training in that basic procedures and the language of the subject domain are expected to be mastered.
Transitional	Represents formal and informal on-the-job type training that builds on entry-level training and places the training in an operational context.
Journeyman	Represents full performance level training, including training necessary to continue full performance.

Kinds of Experience: At issue is the kind of experience brought to the situation that can be transferred or may result in a negative transfer.

None	Any experience would best be represented by basic training.
Low Transfer	Experience in situations different from the current one based on level of war or theater.
High Transfer	Experience in situations the same or similar to the current one based on level of war or theater.

Response Variables

These can be physical (e.g., body strength, sensory deficiencies, or motor skills ability), physiological (e.g., stress, fatigue, or illness), and psychological (e.g., motivation,

intellectual skills or mental state). At the level of resolution of the model, the major concern is how the capacity to respond may be affected. While numerous independent variables can be identified at a high level of resolution, we have chosen to interpret these variables in terms of an intervening variable.

Capacity to Respond:

No Effect	Whatever personal variables are present is not expected to affect performance.
Minimally Decreased	While the capacity to respond is decreased, it would not be expected to cause much difficulty in responding. An example might be boredom that results in a transitory lapse into day dreaming.
Moderately Decreased	The capacity to respond is significantly affected. Examples might be long time periods without sleep, the effect of depressants, or the death of a close friend.

Performance Variables

Performance or operational variables are outside the operator and define the conditions during which the operator must perform. Within each class of operational variables are categories of the variable, which occasion different errors. For example, if time to perform is constrained, we would expect different kinds of errors to be possible than when time to perform is unconstrained. In addition, operational variables are viewed as independently occasioning errors. That is, one category of operational variables does not trigger other categories of operational variables.

The operational variables are identified, based on a particular battlefield environment, the enemy and friendly mode of conducting warfare, and the sensor complement of the BLUEFORCE. As a result, the operational variables are described at a very low level of resolution and represent a composite of the situation rather than the specifics of a high-resolution taxonomy. There are three classes of operational variables: environmental, management, and performance.

Environmental Variables

The environmental variables describe the general conditions in which the tasks are performed. They include variables relating to the immediate environment (e.g., within the work area) as well as the surrounding environment (e.g., within the command post).

Physical Constraints: Any variable that physically limits the human in performing the required tasks. Examples of these are MOPP gear, having to work in a constrained work area such as a van or high mobility multi-purpose wheeled vehicle (HMMWV).

High Level	When constraint makes movement of even gross motor behavior difficult.
Moderate	When constraint makes movement cramped but possible with minimum effort.
Minimal	When constraint does not require much effort.

Ambient Conditions: Any variable that can impose sensory overload on the human. This includes any stimulus condition involving heat, cold, noise, glare, and so forth, that is regarded outside the normal range of acceptance.

Severe	Even the appropriate protective equipment or procedures are only partially effective.
Moderate	Protective gear or procedures are effective if used.
Mild	The sensory conditions are regarded as a minor annoyance.

Management Variables

Management variables include the supervisory, management, and policy controls that impact performance. Some of these variables are dynamic in that they involve the face-to-face and day-to-day operations. Examples of dynamic variables are priorities and suspenses, feedback, reinforcement, and direction and guidance. Other variables are fixed in that they involve written policy and standards that guide or direct behavior. Examples of fixed variables are SOPs, delegations of authority, and doctrine. These variables can have the effect of creating standardization when none is needed and chaos or uncertainty when standardization is appropriate. The different levels of the management variables are not meant to have a good-bad connotation. They trigger different kinds of possible errors.

Management Style

Management style describes how the day-to-day operations are conducted.

Rigid	Operations are "by the book," without deviation. Flexibility is not permitted even when appropriate. The most frequent management responses are direct orders and

	punishment for not going by the book. The goals tend to be determined by the rules rather than the situation. Everything is high priority.
Standardized	While operations are standardized, flexibility when necessary is permitted and encouraged. The most frequent management responses are positive reinforcement for appropriate behaviors and guidance with the intent to train when behavior is inappropriate. The goals are defined by the situation. Priorities are determined, based on the goals and resources available. If the priorities are imposed by external sources, goals and resources are changed to meet the priorities.
Laissez-faire	Uses reinforcement, feedback, and punishment randomly and without respect to the appropriateness of the behavior. Goals are determined by each individual. When priorities exist, they are imposed by external sources and are normally ignored.

Formal Controls

The degree of formalization in the structure of the management control system.

Formal	Policies and procedures are well documented and communicated to everyone. They are readily available for reference.
Available	While policies and procedures are well documented, the individual is responsible for learning and implementing them.
Verbal	Policies and procedures are mostly verbal and subject to frequent unannounced change.
None	For all practical purposes, policies and procedures do not exist.

Performance Control Variables

Performance control variables are those independent variables that control how a task will be performed.

Temporal Constraint

Normally, tasks performed within some time frame as determined by suspenses, priorities, or operating procedures. In addition, the tasks take time to perform. The temporal

constraint is the difference between the time it takes to perform a task to the time available for the completion of the task.

Too Little	Time to perform is less than the time required to complete the tasks. In this situation, for example, the suspense might be met by short cutting the required routine.
Sufficient	Time to perform is time enough to complete the tasks. In this situation, there is no time constraint, but there is also no slack time.
Too Much	Time to perform is more than adequate to the task completion. In this situation, there is enough slack time that several different tasks could be accomplished if necessary.

Performance Criteria

Performance criteria determine how well the task must be done. Usually, performance criteria specify some accepted degree of tolerance. The criteria can be expressed quantitatively, qualitatively, or both.

Specific	Performance criteria are specific, and deviations are unacceptable. For example, if a weapon system requires 8-digit coordinates in order to hit a target, anything less would be useless.
Ranges	Performance criteria exist as ranges of acceptability.
Vague	Performance criteria are vague or nonexistent.

TASK REQUIREMENTS AND JOB AID VARIABLES

These are independent variables that describe the tasks in terms of their resource and support requirements and characteristics and may be used to represent changes in the operational environment to be modeled. These are optional and selectively set by the analyst during model setup. Trigger variable rules may affect these settings; for example, if there is no software for a task, then there also cannot be soft copy, graphics, or symbology.

Job Aids

These are supports that contribute to making task performance easier or more efficient. They must be purposely used by the performer, <u>not</u> transparent. The lack of the job aid does

not prevent the task from being accomplished. For example, if one must sense the enemy's use of poison gas, some kind of sensor is used; the sensor is not a job aid since it is required to accomplish the task.

Procedural Guides	SOPs, letters of instruction, specific guidance contained in operations orders (OPORDs), operation plans (OPLANs), or other documents that describe "how to" perform or implement a function or sub-function. This variable specifies that adequate procedural guides are present to perform a function or sub-function.
References	References, tables, charts, manuals, maps used in the performance of functions or sub-functions. This variable specifies that adequate references are present.
Templates	Templates are job aids prepared before a function is performed, which coalesce an idea or doctrine into a chart or visual aid, thus making analysis and comparison to a norm easier. This variable indicates that templates adequate for the performance of the function or sub-function are present.
Computational Devices	Computational devices are devices such as a calculator which are necessary to perform a function or sub-function. They are normally used to perform mathematical calculations, not to process information. This variable indicates that adequate computational devices are present to perform the function or sub-function.
Specific Software Applications	Software application programs include the use of automation to record, correlate, and extract information or data in support of a function or sub-function. This variable indicates that adequate automated tools are available to perform the function or sub-function.

Stimulus Characteristics

Stimulus characteristics determine response requirements. This category examines characteristics of input into the function, which aid or detract from the analyst's ability to perform a function.

Hard Copy Visual	This variable specifies that paper, photograph, overlays, and so forth, are used in performing the function or sub-function.

Soft Copy Visual	This variable is tied to software application programs. Soft copy includes inputs in the form of "down links" from systems into a computer for data or soft copy imagery and implies computer-to-computer interface for analysis rather than passing paper copies of data or information. Both hard copy and soft copy can be selected for the same function.
Rate of Data Presentation	Use of this variable is to specify a high rate of presentation for a particular function or sub-function. If the rate of data presentation is considered manageable for the function or sub-function, then this variable should not be used.
Symbology	Use of this variable indicates that a system of symbols is used in the performance of the function or sub-function.
Code	This is primarily a collection function rather than a function of processing. It will most likely not be used. For now, code input to the process remains as a place holder for when pre-processing occurs as a part of the intelligence processing functions and not a part of single source analysis performed by collection units.
Waveform	Functions the same as code in that it is a placeholder for future use. Waveform is a mathematical representation of a wave or a graphic deviation at a fixed point versus time. Few processing functions include the use of waveform representation of information.
Graphics	These include the use of overlays and sketches to represent or replace words. They can also be used to enhance verbiage. Most intelligence processing functions include the use of graphic representations of information.
Foreign Language	Input in the form of foreign language is rare. Usually, analysts receive information in a translated form since most linguists are assigned to collection rather than processing functions. This variable should be used when an organization has linguists translating from a foreign language to perform the processing function or sub-function.
Noise Level	This variable is used to specify that the noise level in performing a function is a distracter from performing optimally. It can include physical noise in the surrounding area that causes an analyst to receive less than clear input or distorted signals in receiving input to the function. This variable generally describes a situation when input arrives

predominantly from radio signals and the signal is normally unclear.

Procedural Requirements

Procedural requirements describe how a function or sub-function must be conducted.

Sequential	The sub-functions of the task or function should be performed in order. Failure to perform the sub-functions in order may result in making false assumptions or deductions. The setting is made by determining if the steps of a function are performed sequentially during normal circumstances.
Non-sequential	The order of performing sub-functions of a task is not necessarily important to the performance of the function. All nodes must be either sequential or non-sequential. This cannot be neither or both.
Frequent Shifting Between Tasks	This variable captures interruptions in task performance. It probably occurs more often than not, especially in dynamic situations. If, during the performance of a function, the people performing that function must divert their attention to other areas or functions at the same time, frequent task shifting would be present.
Sustained Attention	There are two areas that must be considered in selecting this variable. First, does the task truly require the sustained attention of the performer? Then, is the performer afforded the time and ability to perform the function with few interruptions routinely? If both of these questions can be answered yes, then this is the proper setting. This setting would be selected for such tasks as administrative logging of data or requirements (perhaps war gaming if a war gaming session is a formalized process with dedicated time and assets). Only one variable, **sustained attention** or **frequent shifting**, can be selected. Selecting neither is not an option.
Group Interaction	Group interaction is the performance of a function or sub-function as part of a group (two or more people) rather than by a single person. Many MI functions are performed by groups rather than individually; for instance, terrain analysis, course of action selection, and war gaming are almost always performed in group work sessions.

Individual Performance	This trigger variable is set for those tasks normally performed by only one person. If group interaction is not selected for a function, then individual performance must be selected and vice versa.

Response Requirements

Response requirements are the mental or physical behaviors required to perform a function or that cause the analysts to respond in certain ways in order to perform a function. These response requirements are at a general level. While any task usually requires a combination of responses, one or two response requirements probably dominate. There are four categories of response requirements with different internal settings: perceptual, motor, cognitive, and communicative. Multiple selections can be made in the same categories.

Perceptual Responses

Visual	In the performance of the function, the analyst must prepare or use visual products. These visuals can be merely textual or may be graphical, as well.
Auditory	The dominant response is through hearing or listening. An example would be extracting data and information about the enemy while reports are being transmitted over a tactical radio system.

Cognitive Responses

Recall	The function requires the analyst to recall information from memory or from a database.
Analyze	This is breaking material into its parts. Many of the intelligence processing functions include analyzing. Make this selection only if analyzing dominates this function.
Integrate/Synthesize	The function or sub-function requires the analyst to create a whole concept by correlating together parts.
Evaluate	The function or sub-function requires the analyst to judge the value of something using criteria. This response is associated strongly with the functions of prioritizing requirements and evaluating the worth of information to be used in processing or selecting the best collection assets.

Motor Responses

Gross Motor	Gross motor skills are skills that require little specialized training. They include drawing, drafting, and writing. They also include physical labor such as heavy lifting, running, marching, and swimming.
Fine Motor	Fine motor skills include fine tuning equipment, steering, and so forth, generally those physical tasks with low tolerance for error. These are seldom required in the intelligence production process.

Communicative Responses

Verbal	The function calls for a verbal response only. This is an informal and untraceable response to a question(s).
Written	The function calls for a more formal or recorded response. The response can be in the form of a written product, graphic, or even a briefing. Although briefings are presented verbally, they normally require preparation of briefing notes and graphics. These contents are usually more duplicable than verbally answering a question. Written response can also include updating a database.

ERRORS

The error framework defines errors as human behaviors that result in deficient outcomes that are, in the MI domain, deficiencies in intelligence. Errors are classified into general and sub-categories as described next. Definitions of specific errors within these types follow.

Types of Errors

General Procedural Errors	Errors that occur when a person is executing procedures.
General Process Errors	Errors that occur when a person is involved in a mental process.
Special Case Administrative Errors	Errors pertaining to compliance or non-compliance with administrative procedures and information that exists to constrain, direct, or guide behavior.
Special Case Information Collection Errors	Errors pertaining to compliance or non-compliance with administrative procedures and information that exists to constrain, direct, or guide behavior.

Errors of Commission	Any type error in which data or information was used improperly or requirements were not properly executed.
Errors of Omission	Any type error in which data or information was not used or considered or requirements were not executed.

APO - Administrative procedural errors of omission.

APO 1 Did not consider the existing administrative constraints, direction, or guidance.

APO 2 Did not consider all the necessary administrative constraints, direction, or guidance.

APRC - Administrative process errors of commission.

APRC 1 Misinterpreted the administrative constraints, direction, or guidance.

CPC - Collecting information procedural errors of commission.

CPC 1 Collected more data than were required to perform the task.

CPC 2 Collected inappropriate data. There can be levels of inappropriate data, that is, all the data were inappropriate or only a few pieces were inappropriate.

CPC 3 Did not collect all the data necessary to perform the task.

CPC 4 Recording or reporting a signal or signal change when none has occurred.

CPC 5 Recording or reporting a signal or signal change in the wrong direction.

CPC 6 Recording or reporting a target when none is in the field.

CPC 7 Assignment of the target to the wrong class.

CPC 8 Responding to a sub-threshold target change.

CPC 9 Premature response to a target change.

CPC 10 Late response to a target change.

CPO - Collecting information procedural errors of omission.

CPO 1 Failed to monitor the field.

CPO 2 Failure to record or report a signal or signal change.

CPO 3 Failure to record or report the appearance of a target.

CPO 4 Failure to respond to a super-threshold target change.

CPRC - Collecting recalling process errors of commission.

CPRC 1 Recalled more information than was necessary to perform the task.

CPRC 2 Recalled inappropriate information. There can be various levels of inappropriate information.

CPRC 3 Did not recall all the information required to perform the task.

CPRO - Collecting recalling process errors of omission.

CPRO 1 Did not recall any information. A case when the person responded reflexively to the environment.

EPC

EPC 1 Inadequate magnitude of control actions.

EPC 2 Excessive magnitude of control actions.

EPC 3 Inadequate continuance of control actions.

EPC 4 Excessive continuance of control actions.

EPC 5 Wrong direction of control actions.

GPC - General procedure errors of commission.

GPC 1 Perform the step(s) incorrectly.

GPC 2 Repeat a step when it is not required to do so.

GPC 3 Insert an unnecessary step.

GPC 4 Perform the steps in the wrong order.

GPC 5 Perform a step before there is enough information to justify doing it.

GPC 6 Perform a step too late.

GPC 7 Perform a step that is similar or unrelated to the required one.

GPO - General procedure errors of omission.

GPO 1 Omit a required step.

GPO 2 Stop the procedure before completing all the steps.

GPRC - General process errors of commission.

GPRC 1 Misinterpreted the information being acted upon.

GPRC 2 Gave information more importance than necessary.

GPRC 3 Failed to keep track of sequential reasoning.

GPRC 4 Lost sight of the reason for performing analysis.

GPRO - General process errors of omission.

GPRO 1 Only used part of the information that is required to perform the step.

GPRO 2 Did not reinterpret existing information in light of new findings.

GPRO 3 Did not integrate new information with existing information.

GPRO 4 Did not associate information from different subject domains.

GPRO 5 Did not build models of events from a mix of hypothesis and facts.

GPRO 6 Did not give information as much importance as necessary.

GPRO 7 Did not reweigh the importance of information based on new information.

HPPRC - Hypothesis procedural or process errors of commission.

HPPRC 1 Used incorrect information to verify or refute predictions.

HPPRC 2 Rejected hypotheses without fully testing the predictions.

HPPRC 3 Accepted hypotheses without fully testing the predictions.

HPPRC 4 Tested hypotheses to a point of diminishing return.

HPPRC 5 Selected a hypothesis having no relationship to current or future possible friendly force or opposing force operations.

HPPRC 6 The hypotheses selected were not supported by the existing information.

HPPRO - Hypothesis procedural or process errors of omission.

HPPRO 1 Did not test any hypotheses.

HPRC - Hypothesis process errors of commission.

HPRC 1 Misinterpreted the information used to verify or refute the hypotheses.

INFORMATION STATE DIMENSIONS

Information is represented by five information state dimensions relevant to intelligence, which provide a non-domain content description of information. The information output by a given node is characterized by the level of the relevant dimensions (not all dimensions are applicable to all nodes).

Relevance

The meaning that is provided to the output by forming relationships within and between various kinds of information.

Fully Relevant	Output contained the appropriate meaning(s).
Mostly Relevant	Output contained most of the appropriate meaning(s).
Limited/Adequate	The meaning in the output was correct, but not all aspects of meaning were covered.
Limited/Insufficient	Output meaning was partially correct, and not all aspects of meaning were covered.
No Relevance	The output was not given meaning.
Wrong Relevance	The output has the wrong meaning.

Specificity

The degree to which output conveys meaning without further interpretation or inference.

Precise	The output is not open to further interpretation or inference.
Precise With Additional Analysis	The output contains little room for further interpretation or inference; interpretation or inference is rather obvious.
Approximate/ Useful	The output contains some room for further interpretation or inference.
Approximate With Major Gaps	The output contains considerable room for further interpretation or inference, so much so that it may be confusing.
Ambiguous	The output is open to different meaning.
Cryptic	The meaning of the output is obscure or concealed.

Completeness

The measure of expected content that should be produced by the function.

All	All inclusive.
Most	The output has most of the expected content.
Some/Sufficient	The output has less than the expected content but is sufficient to work with.
Marginal	The output has less than the expected content; probably insufficient to work with.
Insufficient	The output has none of the expected content.

Perishability

The degree to which output retains its temporal relevance. (Temporal relevance for different tasks is different. The measurement is relative to the function, not relative to functional comparisons.)

Lasting	The output retains its relevance over an extended period of time (full life of the OPLAN).
Temporary/Little Impact	The output retains its temporal relevance for most of the life of the OPLAN.
Temporary/ Adequate	The output retains its temporal relevance for a limited amount of time; sufficient for its intended purpose.
Transit/Some Utility	The output rapidly loses its temporal relevance; may be sufficient for its intended purpose.
Transient/Little Utility	The output rapidly loses its temporal relevance; probably insufficient for its intended purpose.
Elapsed	The output has lost its temporal relevance.

Validity

The soundness of the output as supported by current facts, doctrine, and historical records.

Fully Substantiated	The output is based on pertinent confirmed data.

Mostly Substantiated	The output is based on pertinent, mostly confirmed data.
Partially Substantiated/ Sufficient	The output is based on the data available, some of which are pertinent and confirmed, and the rest questionable or not pertinent.
Partially Substantiated/ Insufficient	The output is based on the few data, some of which are pertinent, confirmed, and the rest questionable or not pertinent.
Unsubstantiated	The output is based on conjecture, incorrect, or irrelevant data.

INFORMATION QUALITY MEASURES AND DIMENSIONS

Military intelligence is information describing battle-related circumstances with sufficient detail to convey a dynamic picture of the enemy and the physical environment. Sufficiency of detail is measured in terms of **completeness** and **specificity**. Generalizing about the battlefield, its enemy, and the physical environment can be further described by addressing or including reference to each of the following aspects, or dimensions, of completeness and specificity:

Completeness

The degree to which the domain-free content of information is thoroughly and totally described.

Specificity

The degree to which information conveys meaning without further interpretation or inference.

Behavior

What is happening? As something on the battlefield is discerned and things are occurring, the descriptions should include their complete and specific descriptions. For example, behavior includes identifying a maneuvering force as attacking or defending. It includes stating that a tank is moving or firing and moving. It includes those descriptive references that allow the user to distinguish what is transpiring because of variations in activity or behavior. Behavior can be

imparted implicitly by distinguishing between a tank and an infantry-fighting vehicle and missile versus tube artillery.

Spatial

Where is it with respect to me? What is directed or discerned on the battlefield is routinely portrayed as a measurable position on the ground or in relation to a distance from a known point area. A grid coordinate on a map is the most specific example. A distance from a city or monument conveys relative location. "West of the Rockies" and "in a sector" are broader spatial references. There are ways to completely and specifically convey spatial relationships, given what is being discerned.

Temporal

What are the time factors? They are always embedded within intelligence. They fix things with regard to the present, past, or future. Their absence or presence conveys urgency and suggests degree of threat.

Structural

What are the parts and how do they fit together? The concept of structure acknowledges that military things are part of larger things. One tank is part of a platoon, a platoon part of a company, and so on. As information about structure becomes more complete and specific, shared knowledge becomes richer. For example, if I know there is a Corps as part of a theater on the field, that conveys more than saying "lots of enemy."

Quantitative

How many? Stating completely and specifically "how many" permits discrete discernment of variability and relative strength.

INFORMATION ATTRIBUTES

Data and information are represented in the IPM in terms of their attributes, that is, what the information is about. These attributes are shape, size, quantity, presence, absence, dynamics, parametrics, and human dimension, and are defined next. This feature enables the IPM to model

information without actual content, as in a message or report that might be produced by an intelligence analyst. Information is said to be "contentless."

Collection activities of the assets described by the scenario are modeled in terms of these information attributes; the inherent information-gathering abilities of a particular collector or asset are described in terms of these attributes, as is the entire asset suite being modeled in the particular scenario. An asset is defined by its contribution (i.e., "yes [Y]," "no [N]," "limited [L],") to collected information having attributes of shape, size, and so forth. Then, given the entire asset suite defined for the scenario, the contribution of each is accumulated by the user, and the overall contribution attributable to the scenario is rated for each of the eight attributes as "excellent," "good," "fair," or "poor."

The total contribution of all assets for a single attribute may be evaluated simply in terms of the collection of Ys, Ls, and Ns or may be adjusted for any peculiarities of the scenario. For example, joint surveillance target attack radar system (JSTARS) has the ability to collect "excellent" information about size and quantity, but if the scenario is taking place in a triple canopy jungle, the attribute ratings for size and quantity may be described as only "fair."

Shape

The physical configuration of an entity, that is, a tank truck or a battalion in march formation. Examples of assets that normally can collect information about shapes are an airborne common sensor (ACS) or ground-based common sensor (GBCS).

Size

The physical extent of an entity, that is, a column of vehicles 2 km long. Examples of assets that normally can collect information about size are forward looking infrared (FLIR) and JSTARS.

Quantity

The count of an entity, that is, 10 infantry fighting vehicles (IFVs). Examples of assets that normally can collect information about quantity are JSTARS and interrogators.

Presence

The existence and location of an entity, that is, a tank at NV263478. Examples of assets

that normally can collect information about presence are ACS and HUMINT.

Absence

The lack of existence, that is, nothing detected. Examples of assets that normally can collect information about absence are JSTARS and HUMINT.

Dynamics

The activity of an entity, that is, movement. Examples of assets that normally can collect information about dynamics are ACS and FLIR.

Parametrics

The technical characteristics of an entity, that is, the pulse repetition frequency (PRF) of a radar. Examples of assets that normally can collect information about parametrics are HUMINT and ACS.

Human Dimension

The human characteristics of an entity, that is, state of training or morale. Examples of assets that normally can collect information about the human dimension are FLIR and ground surveillance radar (GSR).

INTELLIGENCE CONCEPTUAL MAP

The ICM is a node structure that represents how information is developed from database to intelligence to meet the commander's requirements. An ICM node represents information about some aspect of the battlefield in terms of its measures (completeness and specificity) and dimensions (behavioral, spatial, temporal, structural, and quantity). Information in particular nodes is successively combined or integrated into ever-richer information about some broader aspect of the battlefield. Essentially, three information hierarchies are represented in the ICM: enemy, friendly, and physical environment. There are also two levels of information in the hierarchy: the database level, which represents discrete bits of information, and the information-to-intelligence level, which represents successively richer information that results from integrating the discrete bits. Each of the nodes in the ICM is defined next; the three-letter code may be used to cross-reference these definitions with the IPM graphical output.

Intelligence Nodes

Current Activity (ACT)	Information about what the enemy has done recently or is doing now; ongoing activity is correlated with information about enemy forces on the battlefield.
Air (AIR)	Information about the disposition, composition, equipment, and location of enemy aerial assets in the area of interest.
Command, Control, Communications (C-3)	Information about the disposition, composition, and equipment, and location of enemy command, control and communication elements in the area of interest.
Capabilities (CAP)	Information about the enemy's ability to execute various actions; information about both strengths and weaknesses is included.
CS-CSS Unit (CSC)	Information about the disposition, composition, equipment, and location of enemy combat service and combat service support elements in the area of interest.
Demographics (DEM)	Information about all aspects of an enemy population.
Disposition (DIS)	Information about the location and position relationships of enemy forces.
Doctrine (DOC)	Information about how an enemy organizes, trains, sustains, and employs military forces.
Echelon (ECH)	Information about the subordination and echelon relationships of enemy elements such as battalions, regiment, divisions, or Corps.
Enemy Future (ENF)	Information about what an enemy will probably do, and when and where he will do it; in other words, the enemy's likely course(s) of action.
Engineer (ENG)	Information about military engineering activities in the area of interest.
Enemy Now (ENN)	All information about the current state of the enemy; a narrative picture of the battlefield.
Equipment (EQP)	Information about the enemy's inventory of equipment and its capabilities.
Fires (FIR)	All information about enemy fires that support maneuver activities.

Forces (FOR)	Information about all aspects of enemy military organizations; composition, strength, location, and disposition information is correlated with ongoing battlefield activity.
Friendly Operational 1 (FR1)	Information about friendly units, soldiers, and equipment.
Friendly Operational 2 (FR2)	Information about friendly units, soldiers, and equipment.
Friendly Operational 3 (FR3)	Information about friendly capabilities and units.
Friendly Future (FRF)	Information about what our own force will probably do, and when and where he will do it; in other words, the friendly likely course(s) of action.
Friendly New (FRN)	All information about the current state of own forces; a narrative picture of the battlefield.
IA Unit (IA)	Information about enemy infantry and armor units.
Intention (INT)	Information about what the enemy wants to accomplish.
Maneuver Units (MAN)	Information about enemy maneuver units.
Maneuver (MNV)	Information about enemy maneuver.
Morale (MOR)	Information about the morale, well-being, and willingness to fight of enemy units, which affects their capability in the area of interest.
Movement (MOV)	Information about enemy movement on the battlefield.
Mission (MSN)	Information about the actions the enemy has taken or is taking, related to goals, objectives, purposes, and levels of effort involved in each.
Nuclear, Biological, Chemical (NBC)	Information about all aspects of an enemy's ability to conduct or defend against NBC operations.
Physical Environment Future (PEF)	Information about the entire physical environment (the battlefield and associated air space) that might be used in future operations by enemy or friendly forces.
Physical Environment Now (PEN)	Information about the physical environment currently in use by enemy and friendly forces.

PE Effects (PRE)	Information about when, where, and how the terrain and weather will affect enemy and friendly soldiers, equipment, and operations on the battlefield.
Rates (RAT)	Information about the speed of enemy movement on the battlefield.
Readiness (REA)	Information about available combat potential of enemy units in the area of interest.
Reconnaissance (REC)	Information about enemy reconnaissance and intelligence activities.
Size (SIZ)	Information related to numbers of soldiers and equipment involved in activities.
Supporting Units (SPT)	Information about enemy units that provide support to maneuver units.
Staging Areas (STG)	Information about geographic areas in which the enemy prepares for maneuver and stocks supplies.
Subordinate Units (SUB)	Information about supporting units that provide combat power to maneuver units.
Supply (SUP)	Information about enemy logistics activities.
Sustainment (SUS)	Information about enemy sustainment units, that is, resupplying or repairing in the area of interest.
Time/Distance (T/D)	Information about movement in terms of time and distances, that is, how long it will take for a unit to get from point A to B.
Tactics (TAC)	Information about how an enemy conducts military operations on the battlefield.
Type PE (TPE)	Information about all aspects of terrain and weather without regard to a specific operation.
Terrain Analysis (TRA)	Information about terrain in the area of potential operations.
Terrain Effects on Equipment (TRE)	Information about the impact of terrain on equipment, soldiers, and operations on the battlefield.
Terrain Situation (TRS)	All information about the specific terrain in use by both the enemy and friendly forces.

Weather Effects on Equipment (WXE)	Information about the impact of weather on equipment, soldiers, and operations on the battlefield.
Weather Conditions (WXN)	Information about weather in the area of potential operations.
Weather Situation (WXS)	All information about current and projected weather in the area of the terrain in use.
Weather Effects on Terrain (WXT)	Information about the impact of weather on the terrain of the battlefield.

Database Nodes

All data about the people, military, and activities of another country or geographic area except those that are related to the physical environment. They are divided into the following segments:

Historical Database	All data that are known before a specific situation dictates a military operation. These data provide the foundation for populating the current segments of the database and are divided into the following elements:
Organization and Equipment (HBO)	All data known about the soldiers, units, and equipment of a geographic area or a potential adversary.
Activity (HBA)	All data known about historic and recent military movements, emissions, and mission activities of a geographic area or a potential adversary.
Population (HBP)	All data known about the individuals, organizations, and groups of a geographic area or a potential adversary.
Current Organization	All data learned about the **soldiers (OES), units and equipment (OEU)**, and **equipment (OEE)** of a geographic area or a potential adversary after a military operation has been initiated.
Current Activity	All data learned about the military **movements (CAM)**, **emissions (CAE)**, and **mission (CAS)** activities of a geographic area or a potential adversary after a military operation has been initiated.
Current Population	All data learned about the **individuals (CPI), organizations (CPO)**, and **groups (CPG)** of a geographic area or a potential adversary after a military operation has been initiated.

Terrain	All data that are known about the **topography (TRT), hydrology (TRH)**, and **features (TRF)** of a geographic area plus the data learned after a military operation has been initiated.
Weather	All data that are known about the **climate (WXC), meteorological (WXM)**, and **light (WXL)** of a geographic area plus those learned after a military operation has been initiated.
Friendly Database (FDB)	All data that are known about our own forces before and after initiation of a military operation.

NO. OF COPIES	ORGANIZATION
2	ADMINISTRATOR DEFENSE TECHNICAL INFO CENTER ATTN DTIC OCP 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	DIRECTOR US ARMY RESEARCH LABORATORY ATTN AMSRL CS AS REC MGMT 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RESEARCH LABORATORY ATTN AMSRL CI LL TECH LIB 2800 POWDER MILL RD ADELPHI MD 207830-1197
1	DIR FOR PERSONNEL TECHNOLOGIES DEPUTY CHIEF OF STAFF PERSONNEL 300 ARMY PENTAGON 2C733 WASHINGTON DC 20310-0300
1	OUSD(A)/DDDR&E(R&A)/E&LS PENTAGON ROOM 3D129 WASHINGTON DC 20301-3080
1	CODE 1142PS OFFICE OF NAVAL RESEARCH 800 N QUINCY STREET ARLINGTON VA 22217-5000
1	COMMANDER US ARMY RESEARCH INSTITUTE ATTN PERI ZT (DR E M JOHNSON) 5001 EISENHOWER AVENUE ALEXANDRIA VA 22333-5600
1	DEPUTY COMMANDING GENERAL ATTN EXS (Q) MARINE CORPS RD&A COMMAND QUANTICO VA 22134
1	HEADQUARTERS USATRADOC ATTN ATCD SP FORT MONROE VA 23651
1	COMMANDER USA OPERATIONAL TEST & EVAL AGENCY ATTN CSTE TSM 4501 FORD AVE ALEXANDRIA VA 22302-1458
1	CHIEF ARMY RESEARCH INSTITUTE AVIATION R&D ACTIVITY ATTN PERI IR FORT RUCKER AL 36362-5354
1	ARI FIELD UNIT FORT KNOX BUILDING 2423 PERI IK FORT KNOX KY 40121-5620
1	COMMANDANT USA ARTILLERY & MISSILE SCHOOL ATTN USAAMS TECH LIBRARY FORT SILL OK 73503
1	STRICOM 12350 RESEARCH PARKWAY ORLANDO FL 32826-3276
1	GOVT PUBLICATIONS LIBRARY 409 WILSON M UNIVERSITY OF MINNESOTA MINNEAPOLIS MN 55455
1	DR RICHARD PEW BBN SYSTEMS AND TECH CORP 10 MOULTON STREET CAMBRIDGE MA 02138
1	PEO INTELLIGENCE & ELECTRONIC WARFARE ATTN AMCPEO IEW VINT HILL FARMS STATION BLDG 197 WARRENTON VA 22186-5115
1	COMMANDANT US ARMY ARMOR SCHOOL ATTN ATSB CDS (MR LIPSCOMB) FT KNOX KY 40121-5215
1	COMMANDER US ARMY SIGNAL CTR & FT GORDON ATTN ATZH CDM FT GORDON GA 30905-5090
1	DIRECTOR AMC-FIELD ASSIST IN SCIENCE & TECHNOLOGY ATTN AMC-FAST (RICHARD FRANSEEN) FT BELVOIR VA 22060-5606

NO. OF COPIES	ORGANIZATION
1	COMMANDER US ARMY FORCES COMMAND ATTN FCDJ SA BLDG 600 AMC FAST SCIENCE ADVISER FT MCPHERSON GA 30330-6000
1	HQ III CORPS & FORT HOOD OFFICE OF THE SCIENCE ADVISER ATTN AFZF CS SA FORT HOOD TX 76544-5056
1	COMMANDER HQ XVIII ABN CORPS & FORT BRAGG OFFICE OF THE SCI ADV BLDG 1-1621 ATTN AFZA GD FAST FORT BRAGG NC 28307-5000
1	SOUTHCOM WASHINGTON FIELD OFC 1919 SOUTH EADS ST SUITE L09 AMC FAST SCIENCE ADVISER ARLINGTON VA 22202
1	HQ US SPECIAL OPERATIONS CMD AMC FAST SCIENCE ADVISER ATTN SOSD MACDILL AIR FORCE BASE TAMPA FL 33608-0442
1	HQ US ARMY EUROPE AND 7TH ARMY ATTN AEAGX SA OFFICE OF THE SCIENCE ADVISER APO AE 09014
1	COMMANDER HQ 21ST THEATER ARMY AREA CMD AMC FAST SCIENCE ADVISER ATTN AERSA APO AE 09263
1	COMMANDER HEADQUARTERS USEUCOM AMC FAST SCIENCE ADVISER UNIT 30400 BOX 138 APO AE 09128
1	HQ 7TH ARMY TRAINING COMMAND UNIT #28130 AMC FAST SCIENCE ADVISER ATTN AETT SA APO AE 09114

NO. OF COPIES	ORGANIZATION
1	COMMANDER HHC SOUTHERN EUROPEAN TASK FORCE ATTN AESE SA BUILDING 98 AMC FAST SCIENCE ADVISER APO AE 09630
1	COMMANDER US ARMY PACIFIC AMC FAST SCIENCE ADVISER ATTN APSA FT SHAFTER HI 96858-5L00
1	COMMANDER US ARMY JAPAN/IX CORPS UNIT 45005 ATTN APAJ SA AMC FAST SCIENCE ADVISERS APO AP 96343-0054
1	DR SEHCHANG HAH DEPT OF BEHAVIORAL SCIENCES & LEADERSHIP BUILDING 601 ROOM 281 US MILITARY ACADEMY WEST POINT NEW YORK 10996-1784
1	ARL HRED AVNC FIELD ELEMENT ATTN AMSRL HR MJ (R ARMSTRONG) PO BOX 620716 BLDG 514 FT RUCKER AL 36362-0716
1	ARL HRED AMCOM FIELD ELEMENT ATTN AMSRL HR MI (D FRANCIS) BUILDING 5678 ROOM S13 REDSTONE ARSENAL AL 35898-5000
1	ARL HRED AMCOM FIELD ELEMENT ATTN ATTN AMSRL HR MO (T COOK) BLDG 5400 RM C242 REDSTONE ARS AL 35898-7290
1	ARL HRED USAADASCH FLD ELEMENT ATTN AMSRL HR ME (K REYNOLDS) ATTN ATSA CD 5800 CARTER ROAD FORT BLISS TX 79916-3802
1	ARL HRED ARDEC FIELD ELEMENT ATTN AMSRL HR MG (R SPINE) BUILDING 333 PICATINNY ARSENAL NJ 07806-5000
1	ARL HRED ARMC FIELD ELEMENT ATTN AMSRL HR MH (C BIRD) BLDG 1002 ROOM 206B FT KNOX KY 40121

NO. OF COPIES	ORGANIZATION
1	ARL HRED CECOM FIELD ELEMENT ATTN AMSRL HR ML (J MARTIN) MYER CENTER RM 2D311 FT MONMOUTH NJ 07703-5630
1	ARL HRED FT BELVOIR FIELD ELEMENT ATTN AMSRL HR MK (P SCHOOL) 10115 GRIDLEY ROAD SUITE 114 FORT BELVOIR VA 22060-5846
1	ARL HRED FT HOOD FIELD ELEMENT ATTN AMSRL HR MV HQ TEXCOM (E SMOOTZ) 91012 STATION AVE ROOM 111 FT HOOD TX 76544-5073
1	ARL HRED FT HUACHUCA FLD ELEMENT ATTN AMSRL HR MY (B KNAPP) GREELY HALL (BLDG 61801 RM 2631) FORT HUACHUCA AZ 85613-5000
1	ARL HRED FLW FIELD ELEMENT ATTN AMSRL HR MZ (A DAVISON)* 3200 ENGINEER LOOP STE 166 FT LEONARD WOOD MO 65473-8929
2	ARL HRED NATICK FIELD ELEMENT ATTN AMSRL HR MQ (M FLETCHER) ATTN SSCNC A (D SEARS) USASSCOM NRDEC BLDG 3 RM 140 NATICK MA 01760-5015
1	ARL HRED OPTEC FIELD ELEMENT ATTN AMSRL HR MR (M HOWELL) OPTEC CSTE OM PARK CENTER IV RM 1040 4501 FORD AVENUE ALEXANDRIA VA 22302-1458
1	ARL HRED SC&FG FIELD ELEMENT ATTN AMSRL HR MS (L BUCKALEW) SIGNAL TOWERS RM 303A FORT GORDON GA 30905-5233
1	ARL HRED STRICOM FIELD ELEMENT ATTN AMSRL HR MT (A GALBAVY) 12350 RESEARCH PARKWAY ORLANDO FL 32826-3276
1	ARL HRED TACOM FIELD ELEMENT ATTN AMSRL HR MU (M SINGAPORE) BLDG 200A 2ND FLOOR WARREN MI 48397-5000

NO. OF COPIES	ORGANIZATION
1	ARL HRED USAFAS FIELD ELEMENT ATTN AMSRL HR MF (L PIERCE) BLDG 3040 RM 220 FORT SILL OK 73503-5600
1	ARL HRED USAIC FIELD ELEMENT ATTN AMSRL HR MW (E REDDEN) BLDG 4 ROOM 332 FT BENNING GA 31905-5400
1	ARL HRED USASOC FIELD ELEMENT ATTN AMSRL HR MN (F MALKIN) HQ USASOC BLDG E2929 FORT BRAGG NC 28310-5000
1	US ARMY RSCH DEV STDZN GP-UK ATTN DR MICHAEL H STRUB PSC 802 BOX 15 FPO AE 09499-1500
	ABERDEEN PROVING GROUND
2	DIRECTOR US ARMY RESEARCH LABORATORY ATTN AMSRL CI LP (TECH LIB) BLDG 305 APG AA
1	LIBRARY ARL BLDG 459 APG-AA
1	ARL HRED ECBC FIELD ELEMENT ATTN AMSRL HR MM (R MCMAHON) BLDG 459 APG-AA
	ABSTRACT ONLY
1	DIRECTOR US ARMY RESEARCH LABORATORY ATTN AMSRL CS EA TP TECH PUB BR 2800 POWDER MILL RD ADELPHI MD 20783-1197

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1999	Final

4. TITLE AND SUBTITLE

Modeling Intelligence Production Performance

5. FUNDING NUMBERS

AMS: 622716.H700011
PR: 1L162716AH70
PE: 6.27.16
Contract No. DAAL01-94-C-0048

6. AUTHOR(S)

McLean, M.B. (BDM International, Inc.); Knapp, B.G. (ARL)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

BDM Federal, Inc.
4067 Enterprise Way
Sierra Vista, AZ 85635

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
Human Research & Engineering Directorate
Aberdeen Proving Ground, MD 21005-5425

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ARL-CR-444

11. SUPPLEMENTARY NOTES

The contracting officer's representative (COR) is Beverly Knapp, U.S. Army Research Laboratory, ATTN: AMSRL-HR-MY, Greely Hall, Bldg 61801, Ft. Huachuca, AZ 85613-5000 (telephone 520-538-4704).

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The objective of this effort was to develop an analysis framework and computer-based tool for simulating and evaluating the impacts of materiel, organizational, and personnel changes in the military intelligence (MI) production system. This tool was designed to assist the MI community in assessing new concepts for meeting commander's intelligence requirements of the future.

A series of representational models was built first: conceptual, performance, and information quality. The Conceptual Model represented intelligence production as a simple input-process-output model, with nodes representing the functions required to produce intelligence and links representing the information flow. The Performance Model specified the behavioral tasks required to produce intelligence, taxonomy of human performance errors associated with the tasks, and the operational, scenario, and environmental variables that affect task performance. Finally, the Intelligence Quality Model quantified the results of information flow activity and linked the impact of task performance variables when operating on the information.

A team of experts in behavioral science, modeling and simulation, and military intelligence built the Intelligence Production Model (IPM). The computer-based IPM was then built by linking these models using a rule-based logic structure and was accessed by a user interface designed to allow analysts to conduct case studies for a wide range of evaluation questions. The IPM runs in a Windows™-based PC environment and is being applied to a number of questions raised by the MI operational community.

14. SUBJECT TERMS

human system information analysis
intelligence information flow analysis
intelligence production system
rule-based modeling

15. NUMBER OF PAGES

145

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	